Vysoká škola báňská – Technická univerzita Ostrava

Fakulta strojní

Katedra automatizační techniky a řízení

Tvorba relačních databázových systémů

Radim Farana

Ostrava - 1999

Recenzovala: Ing. Milena Olivková

© Ing. Radim Farana, CSc., 1999

ISBN 80-7078-706-6

Názvy programových produktů a firem mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Obsah

| Úvod 5 |
|------------------------------|
| 1. Základní pojmy7 |
| 2. Prostředí MS-Access 10 |
| Prostředí programu MS-Access |
| Vlastnosti prostředí 14 |
| Zobrazit14 |
| Obecné15 |
| Odkazy/HTML15 |
| Úpravy/Hledání 15 |
| Klávesnice |
| Datový list15 |
| Tabulky/Dotazy16 |
| Formuláře/Sestavy16 |
| Další volby 16 |

| 3. Analýza problému17 | | | |
|---|--|--|--|
| 4. Tabulky (Lab01) | | | |
| 5. Dotazy (Lab02)29 | | | |
| Výběrový dotaz QProblemyUkoly29 | | | |
| Výběrový dotaz QProblemyDobaPrace | | | |
| Křížový dotaz QProblemyOsoba | | | |
| Vytvářecí dotaz QHotoveVytvorit | | | |
| Přidávací dotaz QHotovePresunout | | | |
| Odstraňovací dotaz QHotoveOdstranit | | | |
| Aktualizační dotaz QDatumOprava | | | |
| Sjednocovací dotaz QProblemyVsechny | | | |
| Definiční dotaz QHotovePrimarniKlic | | | |
| Použití vložených dotazů (QUkolyDlouhe) | | | |
| Použití parametrů (QUkolyVObdobi) | | | |

| 6. Formuláře (Lab03)42 | | | | |
|--|--|--|--|--|
| Tabelární formulář FOsoby42 | | | | |
| Datový list FUkolySub 46 | | | | |
| Sloupcový formulář FUkoly 47 | | | | |
| Formulář FProblemy 48 | | | | |
| Graf FProblemyDobaPrace | | | | |
| Kontingenční tabulka FProblemyOsoba 54 | | | | |
| 7. Sestavy (Lab04) | | | | |
| Sestava RProblemyUkoly 55 | | | | |
| Sestava RProblemyDobaPrace61 | | | | |
| 8. Makra (Lab05) 62 | | | | |
| Řádek nabídek MenuOsoby65 | | | | |
| Řádek nabídek MenuSestavy 67 | | | | |
| 9. Moduly (Lab06) 68 | | | | |
| Datové typy | | | | |
| Řízení běhu programu73 | | | | |
| Výpis struktury úkolů problému | | | | |

| 10. Bezpečnost systému (Lab07) | |
|---------------------------------|--|
| 11. Replikace (Lab08) | |
| 12. Spolupráce aplikací (Lab09) | |
| 13. Výstup dat na web (Lab10) | |
| Literatura | |

Úvod

Tato příručka byla vytvořena pro studenty Fakulty strojní VŠB-TU Ostrava, kteří absolvují předměty z oblasti databázových systémů (jak bakalářského, tak magisterského studia). Přestože se rozsahy předmětů v různých oborech liší, mají společné zaměření na řešení praktických problémů a tvorbu konkrétních databázových aplikací.

Jako vývojové prostředí se již řadu let (ve skutečnosti od vzniku první verze) používá program Microsoft Access. Jeho možnosti odpovídají jak současným požadavkům na uživatelské rozhraní, tak obvyklým možnostem relačních databázových systémů, pracujících na osobních počítačích třídy IBM-PC. Bohužel jeho nároky na systémové prostředky patří mezi nadprůměrné. Proto je také stále provozována řada aplikací vyvinutých v prostředí 16 bitového OS, přesto že již převládá používání 32 bitového OS Windows.

K systému Microsoft Access je k dispozici řada textů různého určení i rozsahu. Většina učebnic je zaměřena na popis možností systému, které jsou více či méně podrobně rozebírány a doplňovány ukázkovými příklady. Tato příručka je určena především pro samostatné studium, zaměřené na zvládnutí

```
N = 0
For Each L In Human.Languages
    N = N + 1
Next
Print N & "* human"
```

tvorby konkrétní aplikace. Proto je v jejím průběhu řešen jeden zvolený příklad od počátku až do konce. Jednotlivé kapitoly odpovídají jednotlivým třídám objektů, které obvykle navazují v linii tabulka - dotaz - formulář - sestava - makro - modul. Vzájemné propojení objektů (řízení úlohy), však vzniká souběžně se vznikem a propojováním jednotlivých objektů, proto se (stejně jako v reálné situaci) bude v nutné míře popisovat tvorba potřebných programových rutin při vzniku jejich potřeby.

Příručka si neklade za cíl úplný popis možností systému a obsahuje také jen krátké seznámení s nejdůležitějšími používanými pojmy. S teoretickým pozadím databázových systémů je možno se seznámit např. v [*Bejček 1992, Brodský aj. 1992, Tsichritzis 1987, Pokorný 1998, Pokorný a Halaška 1995, 1998*], případně v publikacích věnovaných tvorbě informačních systémů [*Molnár 1992, Moos 1993, Král 1998, Pokorný 1992, Straka 1992, Vlček 1994*]. V poslední době je k dispozici také řada zdrojů na webu, např. [*Computer World, Server FS VŠB-TU Ostrava*]. Jak je vidět, většina publikací je spíše staršího data. Přesto se i v současné době objevují specializované publikace, jako [Pokorný 1994, Šimůnek 1999] věnované dotazovacím jazykům. Většina současných publikací je věnována popisu samotných programů. Jejich zaměření na konkrétní verzi programu však značně omezuje jejich použitelnost. To se samozřejmě týká také systému MS-Access. Již k první verzi byla k dispozici řada publikací [Farana a Vojáček 1993, Nesrsta a Malý 1993, Plecháč 1993]. Záhy přišla verze 2.0 a spolu s ní řada publikací. Vzhledem k velkému rozšíření této verze, jsou stále k dispozici jak publikace popisující základní možnosti systému [Bílek 1995, Farana 1995, Plecháč 1995, *Šimek*]. Tvorbě programových rutin a pokročilým technikám obsluhy se věnují speciální publikace, např. [Plecháč 1994], objevují se také sbírky řešených problémů, jako [Farana 1996, Plecháč a Hernady 1995], dnes dostupné také na webu [Server FS VŠB-TU Ostrava].

S přechodem programů na platformu 32 bitového OS došlo k řadě změn v systému MS-Access, současně se také sjednotilo číslování produktů z rodiny Microsoft Office, proto po verzi 2 následuje verze 7, označovaná také jako 95, neboť MS-Access se stal součástí Microsoft Office 95 Professional. Zatímco publikace k nižším verzím jsou využitelné již jen z části, publikace věnované verzi 7 (95) jsou z většiny platné i nadále. Vyšlo také několik rozsáhlých publikací k tomuto programu, často jako překlady z anglických originálů [*Carlberg 1996, Viescas 1997*]. Ve vydavatelstní Microsoft Press vyšel také velmi dobře zpracovaný CD-ROM určený pro pokročilé uživatele [*Microsoft 1996b*].

Verze 7 byla brzy nahrazena další verzí, označovanou jako 97 (součást Microsoft Office 97 Professional), nicméně interní označení verze jako 8, je stále používáno. Opět je k dispozici řada publikací od přehledových [*Fikáček aj. 1997, Plecháč 1996, Steiner a Valentin 1998*] až po velmi rozsáhlé a podrobné [*Microsoft Office 97, Morkes 1998, Mullen 1997, Simpson a Olson 1998, Varner a Leitgeb 1997, Viescas 1998*]. Nejobsáhlejší informace jsou obvykle i nadále k dispozici v anglických publikacích, např. [*Billings aj. 1997*].

Speciální publikace se zabývají spoluprací jednotlivých produktů rodiny MS-Office [*Microsoft Office 97 Visual Basic, Pokorný 1997b*]. Novou oblastí nasazení databází je tvorba dynamických aplikací na webu. I zde jsou k dispozici specializované publikace, jako [*Cornel 1999, Krejčí 1997, Pokorný 1997a*]. S tvorbou webových aplikací souvisí také tvorba klientských aplikací a využití ODBC [*Plecháč a Semetkovský 1996, Plecháč 1997, Salemi 1993*].

Dalším stupněm v databázových aplikacích je využití databázových serverů (SQL serverů) [*Soukup 1998*], a tvorba aplikací [*Vaughn 1998*].

Na závěr krátkého (a jistě neúplného) výčtu dostupných pramenů je dobré si uvědomit, že se databáze využívají především pro tvorbu informačních systémů a zajištění jejich bezpečnosti je nedílnou součástí tvorby systému [*Dobda 1998*].

1. Základní pojmy

Pro reprezentaci dat v databázi používáme vhodný *datový model*. Donedávna byly používány tři modely: hierarchický, síťový a relační. První dva jmenované se dnes používají ve specializovaných aplikacích, zejména model síťový pro své schopnosti rychlého vyhledávání souvisejících informací. Pracují se záznamy, které se spojují do struktur grafů (tvořených množinou uzlů a množinou hran). Hierarchický model vznikl vývojem v druhé polovině 60. let. Síťový datový model byl navržen skupinou odborníků a zveřejněn ve zprávě CODASYL z r. 1972.

Většina aplikací dnes využívá *relační datový model*. Jeho autorem je Dr. Codda, který jej publikoval v časopise CACM v r. 1971.

Vychází se z matematického pojetí *relace* a *relační logiky*. Relace obsahuje prvky, které reprezentují vztahy mezi nějakými objekty. Formální matematická definice říká: "*Jsou-li D*₁, ..., D_n , (*kde n* > 2) *množiny, pak relace nad množinami D*₁, ..., D_n *je libovolná podmnožina kartézského součinu D*₁' ...' D_n ." Jinými slovy relace je množina prvků (d₁, ..., d_n), kde d_i je z D_i, pro každé i od 1 do n. V kartézském součinu D₁ x D₂ x D₃, ... D_n jsou všechny možné *n-tice* (prvky relace), reprezentující vztahy mezi daty z uvedených množin. Z definice relace plyne: prvky relace musí být různé, jednotlivé D_i nemusí být různé. V databázi uloženou podmnožinu kartézského součinu tvoří ty n-tice, které v dané chvíli odpovídají skutečnosti. Nyní k pojmu *atribut relace*. Je to pojmenovaná množina použitá v kartézském součinu, též nazývaná *doména*. Jednotlivé atributy jsou vzájemně identifikovány *jménem atributu*. Spolu s doménami definují jména atributů *schéma relace*.

Důležitou vlastností relací je nedělitelnost komponent jednotlivých n-tic. Tato atomičnost hodnot tvořících komponenty n-tic se tradičně nazývá *první normální forma relace*. To umožňuje elegantně využít aparát matematické logiky, který s pojmem relace úzce souvisí. Logiku lze totiž použít pro vybudování *relačních dotazovacích jazyků*.

Existují samozřejmě i další pravidla, jejichž dodržení usnadňuje práce s relacemi. Tato pravidla se zabývají vztahy mezi jednotlivými atributy.

Druhá normální forma je jistý přechodný článek ke třetí normální formě. Požaduje, aby žádný *neklíčový atribut* nebyl závislý na vlastní podmnožině *klíče*. Současně jde také o zamezení *funkční závislosti* mezi atributy. Klíč relace je přitom podmnožina atributů relace s těmito vlastnostmi:

- 1. jednoznačná identifikace, tj. každá n-tice relace je hodnotami atributů tvořících klíč jednoznačně identifikovatelná.
- 2. neredundance, tj. žádný atribut klíče nemůže být vynechán, aniž by přestalo platit první pravidlo.

Přitom atribut B je *funkčně závislý* na atributu A, pokud v čase existuje ke každé hodnotě atributu B nejvýše jedna hodnota atributu A.

Nejpoužívanější z těchto pravidel je *třetí normální forma relace*. Jde o ošetření všech funkčních závislostí a *transitivních závislostí*, kdy jsou dva atributy na sobě závislé prostřednictvím dalších. Definice třetí normální formy je: "*Nechť A je množina atributů. Schéma relace R (A) je ve třetí normální formě, jestliže pro každou závislost X->C, kde X je podmnožinou A a C je atribut z A, platí jedna ze tří podmínek:*

- 1. C je obsaženo v X,
- 2. X obsahuje klíč schématu R,
- 3. C je prvkem nějakého klíče schématu R."

Jeden ze způsobů, jak dojít ke třetí normální formě, je založen na *dekompozici*. Schémata se (binárně) dekomponují tak dlouho, až jsou ve třetí normální formě. Potřebujeme k tomu ale vždy znát dvě věci: všechny klíče schématu a všechny funkční závislosti.

Boyce-Coddova normální forma je vylepšením třetí normální formy v tom smyslu, že se netrvá na tom, aby se nevhodné závislosti týkaly pouze neklíčových atributů. Definici Boyce-Coddovy normální formy lze vyjádřit takto: "Schéma relace R (A) je v BC normální formě, jestliže pro každou závislost X->C, kde X je podmnožinou A a C je atribut z A, platí jedna ze dvou podmínek:

- 1. C je obsaženo v X,
- 2. X obsahuje klíč schématu R."

 $\check{C}tvrt\acute{a}$ normální forma je založena na jiném druhu závislosti než předchozí. Nejde totiž o funkční závislosti, ale o tzv. multizávislosti. Uvažujme schéma relace R(A). Jsou-li dány dvě množiny atributů C, D z A, pak multizávislost D na C vychází z intuitivní představy, že jedné C-hodnotě se přiřadí několik Dhodnot. Definice čtvrté normální formy vypadá asi takto: "Schéma relace R (A) je ve čtvrté normální formě, jestliže je v Boyce-Coddově normální formě a každá multizávislost nad A je triviální".

V dotazovacích jazycích se místo pojmu relace používá také pojem tabulka. Jaký je vztah mezi těmito pojmy? Pojem tabulka předpokládá jistou podobu relace, tj. její reprezentaci např. na obrazovce, na papíře apod. Tabulka například pracuje s jistým pořadím řádků, narozdíl od relace, kde se o uspořádání n-tic neuvažuje. Tabulka také může obsahovat stejné řádky, což je v rozporu s relací, kde dvě stejné n-tice nejsou povoleny. Pojem tabulka do relačního datového modelu vnesli v počátcích 70. let tvůrci jazyka SEQUEL (později SQL). Tabulka je svojí podobou bližší uživateli, i když mu zakrývá matematickou podstatu věci. V SQL může mít tabulka duplicitní řádky a jednotlivé řádky mohou obsahovat prázdnou hodnotu (NULL). Jde o jisté porušení matematických pravidel relace, ale je to vlastnost v reálných aplikacích používaná. Dotazovací jazyky SQL však používají poněkud jiný aparát než relační. Také atribut relace není přesně totéž co sloupec tabulky. Atribut zahrnuje celou doménu, kdežto sloupec jen hodnoty dané v tabulce.

Pro práci s jednotlivými n-ticemi relace musíme mít k dispozici způsob, jak jednoznačně určit požadovaný prvek. Pro tyto účely by každá tabulka měla obsahovat *primární (vlastní) klíč*. Primární klíč je atribut (množina atributů), jehož hodnota je pro každou n-tici jedinečná. Atribut, který slouží jako odkaz na jinou tabulku a obsahuje tedy primární klíč jiné tabulky, pak nazýváme *cizí klíč*. Pokud se tabulka účastní více vztahů s ostatními tabulkami, může obsahovat více cizích klíčů. Primární klíč je však v každé tabulce vždy jen jeden.

| Tab | 1 | Porovná | ní | noimů | relačního | а | tabulkového m | nodelu |
|--------|----|-----------|-----|--------|------------|---|---------------|--------|
| 1 u.U. | 1. | 1 010 110 | 111 | pojina | rendennito | u | | loueru |

| Relační datový model | Tabulkový model |
|-----------------------|--------------------------------|
| relace | tabulka |
| schéma relace | záhlaví tabulky |
| n-tice | řádek (také záznam) |
| atribut (nepřesně) | sloupec (také pole či položka) |
| arita (stupeň) relace | počet sloupců |
| doména atributu | množina přípustných hodnot ve |
| | sloupci |

Do databáze se mohou dostat jen takové n-tice, které odpovídají vztahům v reálném světě. Jejich kontrola je prováděna na základě definice *integritních omezení*. Ke schématu relace se přidají jistá tvrzení, která omezují široké možnosti vstupu n-tic do relací. Mohou poskytovat podmínky na hodnoty atributů, na vztahy mezi n-ticemi v jedné relaci (*validační podmínky*), ale i složitá tvrzení popisující vztahy mezi n-ticemi více relací (*referenční integrita*). Splnění těchto omezení zajišťují

programy, které kontrolují vstupující data. Jazyky jako SQL dnes v tomto směru nabízejí rozsáhlé možnosti. Systémový inženýr navrhující schémata relací potom řeší rozpor mezi použitím velkého množství integritních omezení, která zajišťují kvalitní naplnění databáze daty a časem, který zabere kontrola vstupních dat.

S relačním databázovým modelem jsou také spojeny pojmy *entita* a *typ entity*. Entita je popsatelný a jednoznačně určitelný objekt s vlastnostmi vyjádřenými parametry (*atributy*). *Typ entity* je skupina entit se stejnými parametry. Jde o nástroje modelování, používané při vytváření tzv. *konceptuálních modelů databáze*. Nejznámějším nástrojem je *E-R model*. V tomto systému modelujeme databázi pomocí *entit* a *relací* (vztahů, *vazeb*) mezi entitami (viz obr. 1). Jak je zřejmé, pojem relace je zde použit ve zcela jiném významu, než v relačním datovém modelu. I přes vnesení řady problémů, je v databázových programech pojem relace běžně používán, a to ve významu vztahu mezi n-ticemi (řádky tabulky, záznamy).



Obr. 1. E-R model

2. Prostředí MS-Access

V souladu se zaměřením příručky se dále budeme zabývat realizací konkrétní databázové aplikace v programu MS-Access. Obvykle je definován jako objektově orientovaný, událostmi řízený relační databázový systém. To znamená, že je používán relační datový model, resp. tabulkový model, činnost systému je realizována obsluhou jednotlivých událostí, které nastávají v reakci na práci uživatele, případně periodicky podle běhu času. Objektová orientace znamená, že je v svstému definována struktura a vzájemné vztahy jednotlivých objektů (tabulky, formuláře), se kterými můžeme pracovat a využívat zejména předdefinované vlastnosti těchto objektů a jednotlivé metody. Rozhodně to neznamená, že by bylo možno definovat datové objekty odpovídající entitám apod. V dalším popisu se budeme držet pojmů používaných v programu i přes jejich odchvlnost od relačního datového modelu apod.

Informace o všech vytvořených objektech jsou ukládány do jednolitého dokumentu (databázový soubor, také databáze). Ve skutečnosti jsou všechny informace ukládány do systémových tabulek, které nejsou uživatelům přístupné. Celý programový systém je tvořen řadou spolupracujících dokumentů (souborů). Jejich oddělení je vedeno snahou snížit nároky na paměť a využívat vždy jen potřebné části systému. Můžeme se setkat s následujícími tvpy dokumentů:

Tab. 2. Typy dokumentů v systému MS-Access



MSAccess.exe - vlastní systém, doplněný řadou DDL knihoven.

| MDB - da k editaci. |
|------------------------|
| |

atabázový dokument, data a obsluha přístupná



LDB - informace o uzamykání záznamů v databázovém dokumentu při víceuživatelském přístupu. Vytváří ho systém a udržuje během práce s daty.



MDW - definice pracovní skupiny, uživatelů a jejich přístupových hesel (také heslo pro šifrování databáze).



MDE - přeložený a zkomprimovaný databázový dokument nepřístupný pro editaci (tzv. Run-time verze).



MDA - doplňkové služby (Add-Ins), realizované v prostředí MS-Access.



MDZ - šablona průvodců pro podporu práce uživatelů (průvodce tvorbou databáze, tabulky, ...).



MDN - šablona prázdné databáze.

Systém MS-Access je realizován jako jednouživatelský a jednoúlohový. To znamená, že každý uživatel musí spustit vlastní instanci programu a to pro každou úlohu samostatně. Samozřejmě je možné z jedné úlohy přistupovat k více databázovým dokumentům.

V prostředí, obvyklém pro nasazování systému MS-Access, může vypadat dislokace jednotlivých dokumentů jak je uvedeno na obr. 2. Data jsou sdílena, zatímco výkonná činnost je rozdělena na lokální počítače uživatelů.



Obr. 2. Příklad nasazení systému MS-Access

Jako každý systém pracující s daty, musí také MS-Access obsahovat alespoň základní bezpečnostní techniky. Patří mezi ně jednak šifrování dat (pomocí systému RC4 firmy RSA). Jako heslo je přitom použit unikátní identifikátor pracovní skupiny, zadaný při vytváření dokumentu MDW. Ochranný faktor tohoto systému odpovídá jeho základním vlastnostem (XOR systém založený na generátoru náhodných čísel s jádrem určeným heslem a délkou hesla 32 bity). Z uvedených skutečností vyplývá nutnost nepřeceňovat kvalitu systému.

Druhou částí zabezpečení je pak definice skupin uživatelů a uživatelů, včetně určení jejich přístupových práv k jednotlivým objektům jak datové, tak obslužné části. Informace o uživatelích a jejich heslech jsou uloženy v souboru MDW, přidělená práva pak přímo u příslušných objektů.

Zabezpečení proti ztrátě dat nebo porušení integritních omezení (zejména referenční integrity) řeší systém svými prostředky. Na rozdíl od skutečných systémů *klient/server* je každý změněný nebo přidaný záznam ukládán ihned po ukončení editace. Přitom samozřejmě proběhne také kontrola integritních omezení. Na straně serveru také není spuštěna žádná aplikace, která by data spravovala. Proto si také musí jednotlivé klientské programy samy předávat informace o práci (zamykání) jednotlivých záznamů pomocí dokumentu LDB. Systém MS-Access tedy pracuje v architektuře *file server*.

Plnohodnotný *transakční přístup* je možný jen v omezené míře v naprogramovaných uživatelských funkcích. V systémech klient/server zakládá klient transakční objekt, pomocí kterého komunikuje s databázovým serverem (SQL server) pomocí transakcí. Všechny operace během zahájené transakce (BeginTrans) jsou registrovány na serveru a musí být klientem buď potvrzeny (Commit) nebo stornovány (Rollback). Doba na provedení fyzického uložení změn se tak zkracuje na minimum a tím se snižuje nebezpečí, že by nebyly provedeny všechny. Současně jsou tyto systémy schopny archivovat údaje o všech transakcích (žurnál, též auditní stopa), takže je možno i zpětně odhalit původce zásahů do systému a tím doložit jeho činnost (zejména pokud poškodila databázi).

V průběhu zpracování se transakce může dostat do jednoho z pěti stavů:

- *aktivní* (stav od počátku provádění transakce),
- *částečně potvrzený* (stav po provedení poslední operace transakce),
- *chybný* (zjištění, že v normálním průběhu transakce nelze pokračovat);
- *zrušený* (po uvedení databáze do stavu před transakcí -Rollback);
- *potvrzený* (po úspěšném zakončení po potvrzení zápisu všech změn do databáze Commit).

Transakce je ukončena jestliže se nachází ve stavu *potvrzená* nebo *zrušená*. Je-li *zrušená* můžeme ji ponechat *zrušenou*, nebo ji znovu restartujeme (často se tak děje automaticky). Z *chybného* stavu transakce přechází po obnovení původního stavu databáze do stavu *zrušená*. I *částečně potvrzená* transakce se může dostat do stavu *chybná* (např. při výpadku systému v době fyzického zápisu změn do databáze). Přestože systém MS-Access nepracuje v architektuře kliet/server, používají se tyto pojmy i při práci s ním. Na klientském počítači jsou po spuštění úlohy automaticky založeny následující objekty:

- Application vlastní systém MS-Access, včetně všech doplňků a průvodců.
- *Workspace* pracovní prostor uživatele s jeho přístupovými právy, k tomu je potřeba se spojit se souborem pracovní skupiny MDW.

Database - obslužná (klientská) aplikace, buď v podobě Runtime MDE nebo zdrojové podobě MDB.

Klientská aplikace se přitom spojí s datovým dokumentem až při požadavku na práci s daty (tabulky jsou vloženy vazbou attach). Data se jeví jako transparentní součást klientské aplikace a mohou být samozřejmě uložena ve více dokumentech, a také mohou pocházet z dokumentů různých formátů. (Velikost databázového dokumentu MDB je omezena na 1 GB).

Prostředí programu MS-Access



Vlastnosti prostředí

Definice základních vlastností systému je k dispozici v nabídce **Nástroje–Možnosti**:

| Možnosti | ? × |
|--|---|
| Zobrazit Obecné Odkaz Datový list Tabulky/Dotazy Form./! Výchozí barvy Písmo: Černá Pozadí: bílá • Čáry mřížky: světlešed • • Výchozí písmo Písmo: • Árial CE • • Šířka: Velikost: • normální • 10 • | Y/HTML Úpravy/Hledání Klávesnice Sestavy Modul Další volby Výchozí zobrazení čar mřížky ✓ ✓ vodorovné ✓ svislé Výchozí šířka sloupce: 2,501cm Výchozí efekt buňky ✓ plochý ✓ vystouplý ✓ vmáčknutý |
| | |
| | OK Storno P <u>o</u> užít |

V programu je nutno tyto vlastnosti (objektu **Application**) nazývat anglicky. Následující výpis programu tak činí, včetně vysvětlení významu nastavitelných hodnot.

Zobrazit

Application.SetOption "Show Status Bar", True Application.SetOption "Show Startup Dialog Box", False Application.SetOption "Show Hidden Objects", False Application.SetOption "Show System Objects", False Application.SetOption "Show Macro Names Column", False Application.SetOption "Show Conditions Column", False *Pozn.: Zobrazení úvodního okna zajišťuje otevření okna s následující nabídkou:*



Tato nabídka pro rychlé otevření již dříve použitých dokumentů či vytvoření nového dokumentu je zjevně určena hlavně pro začínající uživatele. Pokročilé spíše obtěžuje a rychle se jí zbavují.

Obecné

Application.SetOption "Left Margin", 2.5 Application.SetOption "Right Margin", 2.5 Application.SetOption "Top Margin", 2.5 Application.SetOption "Bottom Margin", 2.5 Application.SetOption "Default Database Directory", "." Application.SetOption "New Database Sort Order", 6 '0 - General, 1 - Traditional Spanish, ..., 6 - Czech Application.SetOption "Provide Feedback with Sound", False

Odkazy/HTML

Application.SetOption "Hyperlink Color", 12 '0 - černá, 1 - tmavočervená, 2 - tmavěžlutá, 3 - tmavěžlutá '4 - tmavěmodrá, 5 - fialová, 6 - šedozelená, 7 - šedá '8 - světlešedá 9 - červená, 10 - světlezelená, 11 - žlutá 12 - modrá, 13 - světlefialová, 14 - modrozelená, 15 - bílá Application.SetOption "Followed Hyperlink Color", 5 Application.SetOption "Underline Hyperlinks", True 'Application.SetOption "Show Hyperlink Adresses in Status Bar", True 'Application.SetOption "HTML Template" 'Application.SetOption "Data Source Name" 'Application.SetOption "User Name" 'Application.SetOption "Password" 'Application.SetOption "ActiveX Server URL" 'Application.SetOption "ActiveX Session Timeout" Nastavení uvozená apostrofem Pozn.: nelze provést v programu, ale pouze ručně v okně Možnosti.

Úpravy/Hledání

Application.SetOption "Default Find/Replace Behavior", 0 '0 - Fast Search, 1 - General Search, 2 - Start of Field Search Application.SetOption "Confirm Record Changes", True Application.SetOption "Confirm Document Deletions", True Application.SetOption "Confirm Action Queries", False Application.SetOption "Show Values in Indexed", True Application.SetOption "Show Values in Non-indexed", True Application.SetOption "Show Values in Remote", False Application.SetOption "Show Values Limit", 1000

Klávesnice

Application.SetOption "Move After Enter", 1
'0 - No, 1 - Next Field, 2 - Next Record
Application.SetOption "Arrow Key Behavior", 0
'0 - Next Field, 1 - nexh Character
Application.SetOption "Behavior Entering Field", 0
'0 - All Field, 1 - Goto Start of Field, 2 - Goto End of Field
Application.SetOption "Cursor Stops at First/Last Field", False
Application.SetOption "Key Assignment Macro", "AutoKeys"
Pozn.: Poslední vlastnost není nastavitelná ručně. Do systému se dostala pro zajištění kompatibility s aplikacemi nižších verzí, kde byla přístupná. Určuje jméno makra obsahujícího definici uživatelských horkých kláves (standardně "Automatické klávesy"). Pokud makropříkaz se standardním názvem nefunguje, zřejmě tuto vlastnost některá aplikace změnila.

Datový list

Application.SetOption "Default Font Color", 0 Application.SetOption "Default Background Color", 15 Application.SetOption "Default Gridlines Color", 8 Application.SetOption "Default Font Name", "Arial CE" Application.SetOption "Default Font Weight", 3 Application.SetOption "Default Font Size", 10 Application.SetOption "Default Font Italic", False Application.SetOption "Default Font Underline", False Application.SetOption "Default Gridlines Horizontal", 10 Application.SetOption "Default Gridlines Vertical", 10 Application.SetOption "Default Column Width", 2.5 Application.SetOption "Default Cell Effect", False Application.SetOption "Show Animations", True

Tabulky/Dotazy

Application.SetOption "Default Text Field Size", 50 Application.SetOption "Default Number Field Size", 2 '0 - Byte, 1 - Integer, 2 - Long, 3 - Single, 4 - Double, 5 - Replecable Application.SetOption "Default Field Type", 0 '0 - Text, 1 - Memo, 2 - Number, 3 - Date/Time, 4 - Currency '5 - Counter, 6 - Yes/No, 7 - OLE Object, 8 - Hyperlink Application.SetOption "AutoIndex on Import/Create", "ID;klíč;kód;číslo" Application.SetOption "Show Table Names", True Application.SetOption "Output All Fields", False Application.SetOption "Enable AutoJoin", True Application.SetOption "Run Permissions", 1 '0 - Orner's, 1 - User's

Formuláře/Sestavy

Application.SetOption "Selection Behavior", 0 '0 - Partialy Enclosed, 1 - Fully Enclosed Application.SetOption "Form Template", "Normal" Application.SetOption "Report Template", "Normal" Application.SetOption "Always Use Event Procedures", False

Další volby

Application.SetOption "Default Record Locking", 0 '0 - No Locks, 1 - All Records, 2 - Edited Record Application.SetOption "Default Open Mode for Databases", 0 '0 - Shared, 1 - Exclusive Application.SetOption "Ignore DDE Requests", False Application.SetOption "Enable DDE Refresh", True Application.SetOption "OLE/DDE Timeout (Sec)", 30 Application.SetOption "Number of Update Retries", 2 Application.SetOption "ODBC Refresh Interval (Sec)", 2500 Application.SetOption "Refresh Interval (Sec)", 60 Application.SetOption "Update Retry Interval (Msec)", 250 Application.SetOption "Command-Line Arguments", "" Application.SetOption "Conditional Compilation Arguments", "" Application.SetOption "Project Name", "Adresar" Application.SetOption "Error Trapping", 1 ' 0 - All Errors, 1 - In Class Module, 2 - No Resumed Errors

3. Analýza problému

Tvorba databázového (informačního) systému začíná jeho analýzou, která vyústí do návrhu jednotlivých tabulek, názvů a datových typů položek, validačních podmínek, primárních (vlastních) klíčů, cizích klíčů a propojení tabulek, s tím také referenční integrity. Tyto údaje současně tvoří významnou část dokumentace systému, proto je do systému MS-Access zařazena služba tisku dokumentace k jednotlivým objektům.

Pro podporu návrhu systému bylo vyvinuto několik modelovacích postupů, E-R modely zmíněné v kapitole 1, HIT model aj. V této příručce se jimi zabývat nebudeme, pouze se přiblížíme E-R modelu.

Naším úkolem bude vytvořit databázovou podporu pro systém řešení problémů postavený na metodě, známé jako *mapa mysli*, analytický pavouk či analytický graf. Jedná se o brainstormingovou metodu, která pomáhá k co nejrychlejšímu a přitom efektivnímu odhalení všech úkolů, které musíme vyřešit, abychom vyřešili zadaný problém. Výsledkem analýzy je graf, viz obr. 3, často zpracovaný velmi neformálně.

Samotný seznam úkolů však pro složitější problémy není dostatečný. Úkoly doplníme také zodpovědnou osobou a dobou řešení. Přitom je samozřejmě možné, aby některé části řešení probíhaly paralelně. A tady je místo pro naši aplikaci, která bude pomáhat při sledování plnění jednotlivých úkolů, informovat o aktuálních úkolech apod.



Obr. 3. Problém organizace abiturientského srazu

Samozřejmě požadujeme současné sledování řešení více problémů. Jednotlivé entity se rýsují již z obr. 3. Potíž je se skutečností, že každý úkol se může rozpadat na dílčí úkoly. Systém je tedy popsán stromovou strukturou úkolů a ta je jak známo definována rekurzivně. Omezit a priori počet úrovní dělení úkolů by zřejmě nebylo jednoduché, když nevíme jak složité problémy budeme evidovat.



Obr. 4. Výchozí E-R model

Na obr. 4 je uveden E-R model obsahující čtyři entity a relace (vazby) mezi nimi. Jako ve většině reálných případů, jsou všechny relace typu 1:N. Typ relace je přitom označován jako *kardinalita*. Kardinalita 1:N znamená, že jeden problém obsahuje více různých úkolů, každý úkol více různých podúkolů, tedy jedné n-tici z entity R(A) náleží více n-tic z entity R(B), ale každé n-tici z R(B) náleží nejvýše jedna n-tice z R(A). Zbývající kardinality jsou 1:1, kdy jedné n-tici z entity R(A) náleží nejvýše jedna n-tice z R(B), a M:N, kdy jedné n-tici z R(A) náleží více n-tic z R(B) i naopak.

Pokud budeme požadovat po systému kontrolu referenční integrity, musíme určit, která z entit (reprezentovaných v relačním datovém modelu relacemi, označovanými v programu jako *tabulky*; tento pojem bude, pro odstranění duplicity významu pojmu relace, používán v dalším výkladu) bude nadřízená. Kontrola referenční integrity pak znamená, že do podřízené tabulky patří jen taková n-tice, která náleží existující n-tici nadřízené tabulky.

Aby kontrola byla proveditelná, je třeba určit podle hodnot kterých atributů se má zjišťovat existence souvisejících n-tic. Na straně nadřízené tabulky je zjevně vhodný *primární klíč*, resp. atributy jej tvořící. Na straně podřízené tabulky nám vhodný atribut (skupina atributů) bude chybět a musíme jej za účelem realizace relace vložit, s tím že bude tvořit *cizí klíč*.

V entitě **Osoba** evidujeme jediný atribut, proto bude současně tvořit primární klíč příslušné tabulky. Cizí klíče u podřízených tabulek budou tvořeny atributem **Osoba**. U entity **Problém** je několik atributů, které by mohly tvořit primární klíč (buď samostatně nebo ve skupině). Vždy se ale jedná o textové údaje, zabírající velký paměťový prostor. To by při jejich použití jako cizí klíč v podřízené tabulce znamenalo značný nárůst velikosti dokumentu. V takových případech se často přidává speciální atribut primárního klíče, realizovaný speciálním datovým typem (automatické číslo), označovaný obvykle ID, který zabírá méně místa. V naší úloze použijeme stejný postup, takže přidáme atribut **ID** do entity **Problém**, jemu odpovídající bude atribut **Problem** tvořící cizí klíč v entitě **Úkol**.

Podobně přidáme atribut **ID** do entity **Úkol**, jemu odpovídající atribut v entitě **Podúkol** budiž nazván **IDMaster**. Zde je ale situace složitější. Každý podúkol je současně nutno chápat jako úkol, který se opět může rozpadat na podúkoly. Navíc již bylo uvedeno, že nevíme do jaké hloubky se mohou úkoly dále dělit na podúkoly.

Jestliže ale můžeme na úkoly i podúkoly pohlížet stejným způsobem, znamená to, že můžeme obě entity sloučit do jediného typu. Entity tohoto typu se budou odkazovat na entity téhož typu. I když to možná vypadá poněkud krkolomně, nejedná se o nijak nepřípustnou konstrukci. Pokud budeme požadovat kontrolu referenční integrity mezi úkoly a podúkoly, znamená to, že také n-tice úkolu musí mít odkaz na nějakou existující n-tici. Zvláště výrazně si tento problém uvědomíme při vložení první n-tice. Tento problém vyřešíme tak, že taková n-tice se bude odkazovat sama na sebe, tedy bude platit rovnost ID = IDMaster.

Výsledkem je pak struktura uvedená na obr. 5. Atributy tvořící klíče (primární i cizí) jsou zvýrazněny. S ohledem na co nejmenší nároky na zabíraný prostor je výhodné, aby jak primární tak cizí klíče byly tvořeny jediným atributem a současně aby tento atribut využíval co nejúspornější datový typ.



Obr. 5. Výsledný E-R model

Závěrem provedené analýzy budiž uvedeno, že cvičný příklad byl zvolen za účelem možnosti prezentace co nejvíce (ale ne všech) možností systému MS-Access, včetně nepříliš často využívaných. (Tabulka úkolů bude obsahovat relaci sama do sebe.)

4. Tabulky (Lab01)

Logicky prvním krokem ve tvorbě aplikace je vytvoření databázového souboru (MDB). Systém nám při této příležitosti nabízí využití **Průvodce databází** (MDZ). Součástí instalace MS-Access je přes dvacet takových průvodců (Adresář, Čas a účty, ...). Pro naši úlohu se ale žádný nehodí a proto vytvoříme prázdný databázový soubor, nebo-li prázdnou databázi (Lab01Sol.MDB). Její součástí je několik systémových tabulek (standardně jsou neviditelné), proto zabírá prázdný databázový soubor 56 kB.

Našim prvním úkolem pak bude vytvořit potřebné tabulky. V souladu s E-R modelem na obr. 5 bude třeba definovat tři tabulky.

Po zadání příkazu k vytvoření nové tabulky se otevře nabídka všech možností vytvoření tabulky, viz obr. 6. Na tomto místě, ještě nevyužijeme nabídky *Průvodce tabulkou* (viz obr. 7), který by nám nabídl sestavit tabulku z předdefinovaných položek, obsahujících také obvyklé validační podmínky. Z pochopitelných důvodů si vyžádáme *Návrhové zobrazení* a budeme všechny položky včetně všech jejich vlastností definovat sami.

| Nová tabulka | ? 🗙 |
|--|---|
| Vytvoří novou tabulku v zobrazení datového listu. | Zobrazení datového listu Návrhové zobrazení Průvodce tabulkou Průvodce importem tabulky Průvodce propojením tabulky |
| | OK Storno |

Obr. 6. Možnosti vytvoření nové tabulky

Průvodce tabulkou

Které z uvedených vzorových tabulek chcete použít k vytvoření nové tabulky?

Označte vzorovou tabulku a vyberte vzorová pole, která má nová tabulka obsahovat. Tabulka může obsahovat pole z několika vzorových tabulek. Pokud si polem nejste jisti, vyberte je a pokračujte. Později je můžete snadno odstranit.

| Vzorové tabulky: | Vzorová pole: | Pole v nové tabulce: |
|--|--|---|
| Adresář Kontakty Zákazníci Zaměstnanci Výrobky Objednávky Rozpis objednávek Pro potřebu podniku O Pro osobní potřebu | KódÅdresářovéPoložky Oslovení Jméno DruhéJméno Příjmení Titul Přezdívka Titul NázevFirmy Adresa | ▶ ▶ >> < < |
| | Storno | <u>Z</u> pět <u>D</u> alší > Do <u>k</u> ončit |

Obr. 7. Průvodce vytvořením nové tabulky

Na obr. 8. je ukázka definice Primární klíč I Problemy : Tabulka položek tabulky Problemy. Každá Název pole Datový typ Popis: uvedena ie položka na 입MID automatické číslo Identifikace problému Osoba Odpovědná osoba FieldName samostatném řádku v horní části text Položka (pole) Název problému Nazev text DataType okna. Zde jsou definovány tři tabulky Popis problému Popis Description memo Cena řešení problému Cena měna základní vlastnosti položky: Termin datum/čas Termín vyřešení problému • *Název* (max. 64 znaky bez.! [Hotovo ano/ne Ukončení řešení problému URL hypertextový odkaz WWW stránky řešení problému 1 ', včetně mezer, unikátní OLE obiekt OLE Dokumentace problému v rámci tabulky). Pro případ Vlastnosti pole přechodu na SQL server apod. Obecné Vyhledávání Vlastnosti je vhodné používat raději Velikost pole dlouhé celé číslo aktuální znaky anglické pouze Nové hodnoty přírůstek Název pole může být položky abecedy. číslice a dolní Formát dlouhý nejvýše 64 Titulek Primární klíč znaků včetně mezer. podtržítko. Chcete-li získat

Indexovat

- Datový tvp, viz tab. 3.
- **Popis** sloužící pouze jako komentář pro autora.

Ostatní vlastnosti položky jsou soustředěny v dolní části okna a jejich skladba se liší podle datového tvpu. Jednotlivé datové typy, včetně poznámek k uložení dat jejich pomocí, jsou uvedeny v tab. 3. U českého názvu, který se nastavuje, je uveden také anglický ekvivalent, který je potřeba znát při programové manipulaci s položkou.

| yhledávání 💧 | | Obecné Vyhledá | vání 💧 | |
|------------------------------|--|---|--|---|
| 50 | Field Size | Zobrazit ovládací prvek | Pole se seznamem | Field Size |
| | Format | Typ zdroje řádků | tabulka/dotaz | Row Source Typ |
| | Input Mask | Zdroj řádků | SELECT Osoby.Osoba FROM Osoby | Row Source |
| Odpovědná osoba | Caption | Vázaný sloupec | 1 | Bound Column |
| | Default Value | Počet sloupců | 1 | Column Count |
| lo Is Not Null | Validation Rule | Hlavičky sloupců | ne | Column Heads |
| Odpovědná osoba musí být zao | Validation Text | Šířky sloupců | 5cm | Column Widths |
| ne | Required | Počet řádků seznamu | 8 | List Rows |
| élku ne | Allow Zero Length | Šířka seznamu | 5,503cm | List Width |
| ne | Indexed | Omezit na seznam | ne | Limit To List |
| | yhledávání) 50 Odpovědná osoba lo Is Not Null Odpovědná osoba musí být zao ne élku ne ne | yhledávání Field Size 50 Format Format Input Mask Odpovědná osoba Caption Default Value Default Value Is Not Null Validation Rule Odpovědná osoba musí být zadé Validation Text ne Required Allow Zero Length Indexed | yhledávání Obecné Vyhledá 50 Field Size Zobrazit ovládací prvek Format Typ zdroje řádků Input Mask Zdroj řádků Odpovědná osoba Caption Default Value Počet sloupců Is Not Null Validation Rule Odpovědná osoba musí být zadá Validation Text ne Required ne Allow Zero Length ne Indexed | yhledáváníObecnéVyhledávání50Field SizeZobrazit ovládací prvekPole se seznamemFormatFormatTyp zdroje řádkůtabulka/dotazInput MaskZdroj řádkůSELECT Osoby.Osoba FROM OsobyOdpovědná osobaCaptionVázaný sloupec1Is Not NullValidation RuleHlavičky sloupcůneOdpovědná osoba musí být zadValidation TextŠířky sloupcůScmneRequiredPočet řádků seznamu8élkuneAllow Zero LengthŠířka seznamu5,503cmneIndexedOmezit na seznamne |

informace o názvech

polí, stiskněte klávesu

F1.

ano (bez duplicity)

Definice hodnot nabízených

pro vyplnění položky

Obr. 8. Definice položek tabulky

Tab. 3. Datové typy

| Nastavení | Datový typ | Velikost |
|-------------------|---|--|
| Text (Text) | (Výchozí hodnota) Text nebo kombinace textu a čísel, a také čísla, která | Maximálně 255 znaků nebo délka nastavená vlastností |
| | nevyžadují výpočty, např. telefonní čísla. | Velikost pole (menší z obou hodnot). Program MS-Access |
| | | nerezervuje místo pro nepoužité části textové položky. |
| Memo (Memo) | Dlouhý text nebo kombinace textu a čísel. | Maximálně 65 535 znaků. (Je-li pole zpracováváno |
| | | prostřednictvím objektů pro přístup k datům a jsou zde |
| | | ukládány pouze texty a čísla [nikoli binární data], je velikost |
| | | pole omezena velikostí databáze.) |
| Číslo (Number) | Číselná data používaná v matematických výpočtech. | 1, 2, 4 nebo 8 bajtů (16 bajtů pouze v případě, že je |
| | | vlastnost Velikost pole nastavena na hodnotu Replikační |
| | | identifikátor). |
| Datum/Čas | Hodnoty kalendářního data a času v intervalu let 100 až 9 999. | 8 bajtů. Desetinné číslo, celá část kóduje datum, desetinná |
| (Date/Time) | | $\operatorname{\check{c}as}(\operatorname{plati} 1 \operatorname{den} = 24 \operatorname{hodin}).$ |
| Měna (Currency) | Hodnoty měny a číselná data využívaná v matematických výpočtech | 8 bajtů. |
| | zahrnujících data s max. čtyřmi des. místy. Přesnost výpočtu je na 15 číslic | |
| | nalevo a 4 číslice napravo od desetinné čárky. | |
| Automatické číslo | Jednoznačné sekvenční (s diferencí 1) nebo náhodné číslo. Číslo přiřazuje | 4 bajty (16 bajtů jen v případě, že je vlastnost Velikost pole |
| (Counter) | program MS-Access vždy při přidání nového záznamu do tabulky. Pole | nastavena na hodnotu Replikační identifikátor). |
| | typu Automatické číslo nelze aktualizovat (jen v programu). | |
| Ano/Ne (Yes/No) | Hodnoty Ano a Ne a položky obsahující pouze jednu ze dvou hodnot | 1 bit. |
| | (Ano/Ne, True/False nebo Zapnuto/Vypnuto). | |
| Objekt OLE | Objekt (například tabulka programu Microsoft Excel, dokument programu | Maximálně 1 GB (omezeno volným místem na disku). |
| (OLE Object) | Microsoft Word, grafika, zvuky nebo jiná binární data) propojený nebo | |
| | vložený do tabulky programu Microsoft Access. | |
| Hypertextový | Text nebo kombinace textu a čísel uložená jako text a používaná jako | Každá ze tří částí datového typu Hypertextový odkaz může |
| odkaz | adresa odkazu. Adresa odkazu se může skládat až ze tří částí: | obsahovat až 2 048 znaků. |
| | zobrazený text - text, který je zobrazen v ovládacím prvku. | |
| | adresa - cesta k souboru (cesta UNC) nebo ke stránce (URL). | |
| | podadresa - pozice uvnitř souboru nebo stránky. | |
| | Nejsnazším způsobem jak do pole nebo ovládacího prvku vložit adresu | |
| | odkazu, je klepnout v nabídce <i>Vložit</i> na příkaz <i>Hypertextový odkaz</i> . | |

| Vytvoří položku, která umožňuje vybírat hodnoty z jiné tabulky nebo ze | Stejná velikost jako u položky primárního klíče |
|--|--|
| seznamu hodnot prostřednictvím seznamu nebo pole se seznamem. | používaného k vyhledávání, zpravidla 4 bajty. |
| Klepnutím na tuto volbu se spustí Průvodce vyhledáváním, který vytvoří | |
| vyhledávací pole. Jakmile práci s průvodcem ukončíme, program MS- | |
| Access nastaví datový typ na základě hodnot, které byly v průvodci | |
| vybrány. | |
| | Vytvoří položku, která umožňuje vybírat hodnoty z jiné tabulky nebo ze seznamu hodnot prostřednictvím seznamu nebo pole se seznamem. Klepnutím na tuto volbu se spustí Průvodce vyhledáváním, který vytvoří vyhledávací pole. Jakmile práci s průvodcem ukončíme, program MS- Access nastaví datový typ na základě hodnot, které byly v průvodci vybrány. |

Pozn.: Položka typu Memo, Hypertextový odkaz a Objekt OLE nelze indexovat.

Tab. 4. Číselné datové typy

| Nastavení | Popis | Počet des.míst | Paměťové nároky |
|------------------------------|---|------------------|--------------------|
| Bajt (Byte) | Čísla od 0 do 255 (ne zlomky). | Žádné | 1 bajt |
| Celočíselný (Integer) | Čísla od -32 768 do 32 767 (ne zlomky). | Žádné | 2 bajty |
| Dlouhý celočíselný (Long) | (Výchozí hodnota) Čísla od -2 147 483 648 do 2 147 483 647 (ne zlomky). | Žádné | 4 bajty |
| Jednoduchá přesnost (Single) | Čísla od -3,402823E38 do -1,401298E-45 pro záporné hodnoty a od 1,401298E-45 do 3,402823E38 pro kladné hodnoty. | 7 | 4 bajty |
| Dvojitá přesnost (Double) | Čísla od -1,79769313486231E308 do -4,94065645841247E-324 pro záporné hodnoty a od 1,79769313486231E308 do 4,94065645841247E-324 pro kladné hodnoty. | 15 | 8 bajtů |
| Replikační identifikátor | Globálně jedinečný identifikátor (GUID) | Není k dispozici | 16 bajtů |

Vlastnosti položek jednotlivých datových typů si ukážeme na příkladech. Položka *Osoba* je datového typu text. Na kartě *Obecné* je uvedeno:

- *Velikost pole:* 50 (maximální počet znaků, má smysl výrazně omezovat jen u položek patřících do klíčů),
- *Formát*: (předpis pro zobrazení vloženého textu, podrobně popsán v nápovědě, nepříliš využívaný),
- *Vstupní maska*: (předpis pro vkládání textu, opět popsána v nápovědě; je třeba věnovat pozornost souladu s formátem zobrazení, aby již zapsaný text při opravě neodporoval vstupní masce),
- *Titulek*: Odpovědná osoba (při práci s položkou se zobrazuje v záhlaví a nahrazuje tak jméno položky),
- *Výchozí hodnota*: (hodnota nabízená pro položku nového záznamu),
- *Ověřovací pravidlo*: Is Not Null (validační podmínka, zde položka nesmí být prázdná; kontroluje se po změně obsahu položky a při pokusu o uložení záznamu),
- *Ověřovací text*: Odpovědná osoba musí být zadána (text zobrazený jako hlášení v případě porušení validační podmínky),
- *Je nutno zadat*: ne (volba ano je adekvátní výše zadané validační podmínce Is Not Null. Rozdíl je v tom, že hlášení v případě jejího porušení je systémové.),
- *Povolit nulovou délku*: ne (určuje, zda řetězec nulové délky ("") je platnou hodnotou položky),
- *Indexovat*: ne (určení, zda bude vytvářen klíč tvořený touto položkou; je třeba řádně rozlišovat indexování s povolením duplicity hodnot a bez ní.).

Na kartě *Vyhledávání*, která definuje seznam hodnot, nabízených pro položku, jsou uloženy vlastnosti používané také pro objekty formuláře. Tím poněkud předbíháme běžný postup práce, proto význam jednotlivých vlastností bude popsán jen zběžně. Navíc můžeme využít *Průvodce vyhledáváním* (v nabídce datového typu), který s vyplněním vlastností velmi účinně pomůže.

Zobrazit ovládací prvek: Pole se seznamem (typ objektu formuláře pro zobrazení položky; bude automaticky použit při vložení položky do formuláře či sestavy).

Typ zdroje řádků: tabulka/dotaz (nebo seznam hodnot nebo seznam položek určené tabulky),

Zdroj řádků: SELECT Osoby.Osoba FROM Osoby; (SQL dotaz definující datový zdroj, zde seznam osob z tabulky **Osoby**).

| Vázaný sloupec: | 1 (uloží se hodnota ze sloupce 1), |
|----------------------|------------------------------------|
| Počet sloupců: | 1, |
| Hlavičky sloupců: | ne (nebo ano - zobrazit), |
| Šířky sloupců: | 5cm, |
| Počet řádků seznamu: | 8 (po rozvinutí nabídky), |
| Šířka seznamu: | 5,503cm (0,5 cm na svislý |

posuvník – pokud počet hodnot zdroje řádků přesahuje počet řádků seznamu)

Omezit na seznam: ne (je možno zadat i hodnotu, která není ve zdroji řádků; pokud je vázaný sloupec jiný než první, pak se vždy musí vybrat některá z nabízených hodnot). Pro ostatní položky uveďme jen zajímavé vlastnosti. Pro položku *Cena*:

Ověřovací pravidlo: Is Not Null And >=0,

Ověřovací text: Cena řešení problému musí být stanovena (a nezáporná)

Pro položku *Termin* (termín splnění úkolu):

Výchozí hodnota: =DateAdd("m";1;Date()) (funkce vrací aktuální datum posunuté o jeden měsíc),

Pro položku *Hotovo* (datový typ ano/ne):

Výchozí hodnota: FALSE (je vhodné určit jednu z hodnot TRUE/FALSE neboli ano/ne a zabránit tak v případě nevyplnění hodnotě Null, která se při grafickém zobrazení nedá odlišit od hodnoty FALSE),

Globální vlastnosti tabulky jsou soustředěny v samostatném okně, viz obr. 9. Mezi nejdůležitější patří *Popis*, nahrazující při práci s tabulkou její název (obdobně jako *Titulek* nahrazuje název položky) a ověřovací pravidlo. To je určeno pro validační podmínku, obsahující více položek tabulky (např. Datum ukončení řešení musí být větší než Datum zahájení řešení – [DatumUkonceni]>[DatumZahajeni]), která by se u těchto položek jednotlivě definovala velmi těžko. Názvy položek se přitom uzavírají do hranatých závorek (Stejně tomu bude při jakémkoliv jejich použití dále.)

V okně *Indexy* jsou zobrazeny definice klíčů (primárních i sekundárních neboli vlastních i cizích). Systém sem automaticky přenáší definice klíčů z vlastnosti *Indexovat*

uvedené u položek. Kromě toho je zde možné vytvářet zejména *složené klíče* (obsahující více položek). Při vytváření klíčů je třeba postupovat uvážlivě. Použití klíče při vyhledávání a řazení sice urychluje práci, ale nutnost průběžné aktualizace klíčů ji naopak zpomaluje při editaci záznamů.





| Index Name Field Name | | | | |
|--------------------------|--|--|--|--|
| Sort Order | | | | |
| | | | | |
| Soft Older | | | | |
| | | | | |
| Vlastnosti indexu | | | | |
| Primary | | | | |
| Unique | | | | |
| Ignore Nulls | | | | |
| I U I | | | | |



Stejným postupem vytvoříme také ostatní tabulky, proto nyní uvádíme přehledně pouze nejdůležitější vlastnosti:

| Položka (název) | Datový typ | Výchozí hodnota | Ověřovací pravidlo | Poznámka |
|-----------------|---------------------------|-----------------|--------------------|---------------------------------|
| ID | automatické číslo | | | |
| IDProblem | číslo (dlouhé celé číslo) | 0 | | Odkaz na problém |
| IDMaster | číslo (dlouhé celé číslo) | 0 | | Odkaz na nadřízený úkol |
| Osoba | text (50) | | Is Not Null | Vyhledávání viz Problemy |
| DatumOd | datum/čas | =Date() | Is Not Null | |
| DatumDo | datum/čas | | | |
| Reseni | text (255) | | | |

Tabulka Ukoly (Evidence jednotlivých úkolů)

Ve vlastnostech tabulky nastavíme validační podmínku určující, že datum ukončení musí následovat za datumem začátku (pokud existuje) a můžeme také určit způsob řazení záznamů:

Ověřovací pravidlo: [DatumDo] Is Null Or [DatumDo]>=[DatumOd]

Ověřovací text: Datum ukončení úkolu nemůže předcházet datum začátku řešení

Řadit podle:IDProblem;DatumOd;DatumDo

Tabulka **Osoby** (Seznam řešitelů)

| Položka (název) | Datový typ | Výchozí hodnota | Ověřovací pravidlo | Poznámka |
|-----------------|------------|-----------------|--------------------|----------|
| Osoba | text (50) | | Is Not Null | |

Tato tabulka je zjevně pomocná a slouží pouze k nabízení hodnot pro položky *Osoba* v tabulkách *Ukoly* a *Problemy*. Větší smysl by získala, pokud budeme požadovat evidenci více údajů o řešitelích (kontakt, zaměstnavatel, bydliště, ...).

Pozn.: Vzhledem k jednoduchosti řešené úlohy jsme se při její analýze omezili na **datovou analýzu**, která vedla k návrhu datově optimalizovaných tabulek. Na ni by měla navazovat **funkční analýza**, zkoumající manipulaci s objekty. Zejména v rozsáhlých informačních systémech může ukázat vhodnost duplikování položek, tvorby tabulek agregovaných údajů apod. Protože neustálé manipulace se všemi daty by způsobily neúměrné prodloužení odezvy systému. Blíže např. [Pokorný 1992]

Posledním krokem v této části řešení problému je tvorba vazeb mezi tabulkami (relace podle E-R modelu) a zejména nastavení pravidel *referenční integrity*.

okna *Relace* postupně Do vložíme tabulky Osobv. Problemy a Ukoly. Vazbu mezi tabulkami vytvoříme uchopením vazební položky v nadřízené tabulce a položením na vazební položku v tabulce podřízené. Následně se otevře okno *Relace* (v levé dolní části obr. 11), kde můžeme především nastavit požadavky na automatickou kontrolu referenční integrity (Zajistit referenční integritu) a následně povolit automatické řešení kolizí při manipulaci s nadřízenou tabulkou (Aktualizace souvisejících polí *v kaskádě* – po změně hodnoty nadřízené položky, Odstranění souvisejících polí v kaskádě – po odstranění záznamu v nadřízené tabulce).



Obr. 11. Vazby mezi tabulkami a referenční integrita

Nastavení typu spojení slouží k určení obvyklého způsobu spojování záznamů z obou tabulek při tvorbě SQL dotazu. Jednotlivé volby mají následující význam:

1: INNER JOIN – vnitřní spojení

- 2: LEFT JOIN levé polospojení
- 3: RIGHT JOIN pravé polospojení

Typ vazby (1:1, 1:N) se nenastavuje. Systém ho sám určí podle vlastností vazebních položek, jmenovitě podřízené položky. Pokud je indexována s povolením duplicit, jedná se o vazbu 1:N, pokud bez povolení duplicit, pak o vazbu 1:1. Z toho mimo jiné vyplývá, že pro zajištění automatické kontroly referenční integrity je nutné, aby položka nadřízené tabulky byla indexována bez povolení duplicit! Obvykle to není problém, neboť vazbu ze strany nadřízené tabulky většinou tvoří *primární klíč*.

Pro ostatní vazby uvedeme opět jen důležité vlastnosti:

Osoby-Problemy

- Zajistit referenční integritu
- Aktualizace souvisejících polí v kaskádě

Osoby-Ukoly

- Zajistit referenční integritu
- Aktualizace souvisejících polí v kaskádě

Nejzajímavější bude zajištění automatické kontroly referenční integrity mezi záznamy tabulky *Ukoly*, tedy zajištění, že každý

podúkol patří již existujícímu úkolu. Tento požadavek zajistíme druhým vložením tabulky *Ukoly* do okna *Relace* (bude automaticky označena jako *Ukoly_1*). Nyní vytvoříme vazbu mezi tabulkami *Ukoly* a *Ukoly_1* od nadřízené položky *ID* do podřízené položky *IDMaster*. Následně nastavíme požadavek kontroly referenční integrity.

Pozn.: Při příštím otevření okna Relace systém změní zobrazení vazby tak, že hlavní bude tabulka Ukoly_1 a podřízená tabulka Ukoly, jak je zobrazeno na obr. 11. Na správnosti fungování systému se tím nic nezmění.

Tím je tvorba tabulek ukončena a můžeme zkontrolovat jejich funkčnost, zejména fungování *validačních podmínek* a *kontroly referenční integrity*:

- Není možno vytvořit úkol a neurčit ke kterému problému patří (tedy také vytvořit úkol při neexistenci problému).
- Není možno vytvořit problém pokud správně nevyplníme položku IDMaster (může obsahovat hodnotu ID z téhož záznamu tak budeme poznávat hlavní úkoly, které neobsahují nadřízené úkoly).
- Není možno vytvořit problém (ani úkol), pokud neurčíme odpovědnou osobu, tedy pokud nebude existovat alespoň jeden řešitel.
- Datum splnění úkolu nemůže předcházet datum začátku řešení tohoto úkolu.

5. Dotazy (Lab02)

Dotazy slouží, jak jejich název napovídá, k vykonávání SQL dotazů. K jejich tvorbě můžeme použít jak grafické prostředí (QBE – Query by Example – tvorba dotazu podle vzoru), ale také zápis přímo v jazyce SQL. Vzhledem k intuitivnosti ovládání grafického prostředí budeme tvořit dotazy spíše v něm, a SQL zápis uvedeme spíše pro doplnění.

V prostředí MS-Access jsou k dispozici různé typy dotazů, které odpovídají různým SQL příkazům:

- Výběrový dotaz (SELECT QUERY)
- Křižový dotaz (CROSSTAB QUERY)
- Vytvářecí dotaz (MAKE TABLE QUERY)
- Přidávací dotaz (APPEND QUERY)
- Odstraňovací dotaz (DELETE QUERY)

Do skupiny SQL dotazů jsou navíc zařazeny dotazy, které není možno vytvářet v grafickém prostředí QBE:

- Sjednocovací (UNION) seskupení záznamů z více tabulek do jedné společné
- *Předávací* zasílání SQL příkazů prostřednictvím ODBC (např. do databáze spravované Microsoft SQL Server).
- *Definiční* vytváření či modifikace objektů (tabulek) SQL příkazy (CREATE TABLE, ALTER TABLE, DROP, ...).

Výběrové dotazy provádějí především tři základní relační operace:

- Selekce (výběr požadovaných záznamů),
- Projekce (výběr požadovaných položek),
- Spojení (spojení záznamů z více tabulek).

Kromě toho je možno realizovat:

- pojmenování (přejmenování) položky,
- vytvoření nové (virtuální) položky,
- provádění agregovaných výpočtů nad skupinami záznamů,
- požadované řazení záznamů.

Z uvedeného je zřejmé, že dotazy jsou vybaveny značnými možnostmi, zejména pro hromadné zpracování velkých objemů dat. Většinou se používají pro přípravu dat pro formuláře a sestavy, jako příprava dat pro programové zpracování, export apod.

Výběrový dotaz QProblemyUkoly

Jeho úkolem je především spojit data problémů a úkolů, dále vypočítat dobu práce na úkolu a vybrat jen ty úkoly, které nejsou ukončeny (Hotovo=False)

Na obr. 12 je zobrazeno prostředí QBE pro tvorbu dotazu. V horní části okna se prování spojení tabulek.



Obr. 12. Tvorba výběrového dotazu

Po vložení požadovaných tabulek (*Problemy* a *Ukoly*) se do horní části okna dotazu ihned vloží vazba mezi nimi, definovaná dříve v okně *Relace*. V našem případě je použito levé polospojení (přes rovnost vazebních položek *ID* a *IDProblem*), takže budou zobrazeny také údaje problémů, které nemají definován žádný úkol.

Následně provedeme výběr zobrazených položek (Projekce), přetažením požadovaných položek do jednotlivých sloupců v dolní části. S výhodou využijeme zjednodušeného zadání pro zobrazení všech položek tabulky **Problem** zapsané znakem "*". Z tabulky **Ukoly** vybereme požadované položky jednotlivě. Do dalšího sloupce zapíšeme definici virtuální položky vypočítávající dobu práce na úkolu (pokud je zadána doba ukončení úkolu), uvozené jejím jménem:

DobaPrace:

```
IIf(IsNull([DatumDo]);Date();[DatumDo])
-[DatumOd]
```

Pro výběr požadovaných záznamů vložíme znovu položku *Hotovo* z tabulky *Problemy* a do řádku *Kritéria* zapíšeme podmínku "FALSE". Dvojnásobné zobrazení položky Hotovo odstraníme zrušením požadavku na zobrazení položky.

Obdobně zajistíme požadované řazení záznamů do skupin stejného problému (*ID* z tabulky *Problem*) a dále podle data začátku a konce řešení úkolu, viz obr. 12. Pořadí významnosti řazení je přitom bráno zleva doprava podle polohy položek v definici.

Na závěr určíme ještě společné vlastnosti dotazu, viz obr. 13.

| 💣 Vlastnosti dotazu | X | |
|--------------------------------|---------------------------------|-------------------------|
| Obecné | | |
| Popis | Spojení problémů a jejich úkolů | Description |
| Výstup všech polí | ne | |
| Nejvyšší hodnoty | vše | TOP n (TOP n PERCENT) |
| Jedinečné hodnoty | ne | DISTINCT |
| Jedinečné záznamy | ano | DISTINCTROW |
| Práva pro spuštění | uživatel | WITH OWNERACCESS OPTION |
| Zdrojová databáze | (aktuální) | IN |
| Zdrojový propojovací řetězec . | | |
| Uzamčení záznamů | žádné uzamčení | Record Locks |
| Typ sady záznamů | dynamická sada | Recordset Type |
| Doba odezvy ODBC | 60 | ODBC Timeout |
| Filtr | | Filter |
| Řadit podle | | Order By |
| Maximum záznamů | | Max Records |
| | | |

Obr. 13. Vlastnosti dotazu

Výsledek dotazu je zobrazen jako *Datový list* viz obr. 14, stejně jako tabulka. I přes značnou podporu editace a vkládání dat (nabídka jmen řešitelů apod.) není vhodné zpřístupňovat dotazy uživatelům přímo, ale vždy prostřednictvím formulářů. Proto se dále prací s výsledky dotazů nebudeme zabývat.

Pozn.: Všimněte si přejmenování položky **Osoba** z tabulky **Ukoly**. Odstraňuje se tím duplicitní název položky (**Osoba**), který je nepřípustný. Jinak by problém systém odstranil rozšířením jména položky o název tabulky (**Ukoly.Osoba**). Díky zobrazení **Titulku** v záhlaví položky bychom si této skutečnosti nevšimli.



Obr. 14. Výsledek dotazu (virtuální tabulka)

Výběrový dotaz QProblemyDobaPrace

Dotaz zpracuje celkové součty odpracované doby na jednotlivých problémech. Základem bude dotaz *QProblemyUkoly*. Z něj vybereme požadované položky (*ID*, *Osoba*, *Nazev*, *DobaPrace*, *Hotovo*). V nabídce *Zobrazit* si vyžádáme *Souhrny*, což způsobí přidání další vlastnosti pro položky. Položky *ID*, *Osoba*, *Nazev*, použijeme pro seskupení záznamů do jednotlivých skupin. Zde je nutné zajistit, aby mezi těmito položkami byl také primární klíč, který zajistí správné rozlišení skupin úkolů skutečně pro jednotlivé problémy. Položka *DobaPrace* bude agregována funkcí *Sum* (součet). Poslední položka *Hotovo* slouží pro selekci záznamů (nastavení *Kde*). Výsledná SQL podoba dotazu je:

SELECT DISTINCTROW QProblemyUkoly.ID, QProblemyUkoly.Osoba, QProblemyUkoly.Nazev, Sum(QProblemyUkoly.DobaPrace) AS DobaPrace FROM QProblemyUkoly

WHERE (((QProblemyUkoly.Hotovo)=False))

GROUP BY QProblemyUkoly.ID,

QProblemyUkoly.Osoba, QProblemyUkoly.Nazev;

Pozn.: Všimněte si přejmenování položky **DobaPrace**. Systém by agregovanou položku automaticky přejmenoval na SumOfDobaPrace, tedy složením agregační funkce a původního názvu položky. Odstraňuje se tím možnost vzniku duplicitního názvu položky, pokud by stejná položka byla použita také pro tvorbu seskupení.

Křížový dotaz QProblemyOsoba

Křížový dotaz provádí přehledové zpracování dat (agregaci dat jedné položky v závislosti na dvou podmínkách). V našem případě to bude určení celkové doby řešení úkolů pro jednotlivé problémy (po řádcích), rozdělená podle jednotlivých řešitelů (jejich údaje budou zařazeny do sloupců).

I když tento problém vypadá složitě, jeho řešení bude jednoduché díky použití speciálního *Průvodce křížovým dotazem*. V pěti krocích postupně určíme:

- 1. Zdrojový objekt obsahující všechny potřebné položky (*QProblemyUkoly*).
- Položky určující rozdělení záznamů do skupin pro jednotlivé řádky – *Hlavičky řádků (ID, Osoba, Nazev*). Mohou být nejvýše tři a musí mezi nimi být položka jednoznačně určující problém, tedy primární klíč tabulky *Problemy*. Více položek je možno určit pouze ručně.
- Položku určující rozdělení záznamů do jednotlivých sloupců – *Hlavičky sloupců* (*Resitel*). Taková položka může být pouze jedna.
- 4. Položku pro vyplnění vzniklé tabulky (*DobaPrace*) a současně agregační funkci, která se použije pro zpracování všech záznamů padajících do jednoho pole tabulky (Sum). Současně můžeme požádat o vytvoření součtů všech údajů v jednom řádku. Tím jsou provedeny všechny důležité úkony, viz obr. 15.
- 5. Nakonec určíme název výsledného dotazu.

Průvodce křížovým dotazem

| Jaké číslo se má sp průsečík řádku a sl Můžete například v Velikost objednávky zaměstnace (sloupe nebo regionu (řádel Chcete sčítat každj I Ano, zahrnout s | oočítat pro každý oupce? y pro každého ec) podle země k). ý řádek? oučty řádků. | Pole: Cena Termin Hotovo Datum0 Datum0 Reseni DobaPro | d o ace | Fu C F L M S S V | inkce: vg count irst ast fax fax tin tDev <u>um</u> |
|--|---|--|----------------|---------------------------------------|--|
| Příklad: | | | | | |
| ID | Osoba | Nazev | Resitel1 | Resitel2 | Resitel3 |
| ID1 | Osoba1 | Nazev1 | Sum(DobaPrace | ;) | |
| ID2 | Osoba2 | Nazev2 | 1 | | |
| ID3 | Osoba3 | Nazev3 | 1 | | |
| ID4 | Osoba4 | Nazev4 | 1 | | |
| | | | | | |
| | | Storno | < <u>Z</u> pět | <u>D</u> alší > | Do <u>k</u> ončit |

Obr. 15. Průvodce křížovým dotazem, krok 4

Pozn.: Výsledkem dotazu je tabulka, obsahující všechny sloupce určené jako hlavičky řádků a dále tolik sloupců, kolik bylo různých hodnot v položce určené pro hlavičky sloupců. Pokud se mezi nimi vyskytovala hodnota NULL, bude nahrazena "<>". Protože je výsledkem dotazu tabulka a tu není vhodné zpřístupňovat uživateli, používá se spíše formulář využívající kontingenční tabulku z MS-Excel.

| 🛱 QProblemyO | lsoba : Křížový do | ıtaz | | | | |
|--|--------------------|----------------|----------------|------------------|--------------------|------------------------------|
| Problet * ID Osoba Nazev Popis | nyUkoly | | | | Realiza | ace součtu hodnot v řádku |
| | | | | | | |
| Pole: | ID | Osoba | Nazev | Resitel | Hodnota: DobaPrace | DobaPraceCelkem: DobaPrace |
| Tabulka: | QProblemyUkoly | QProblemyUkoly | QProblemyUkoly | QProblemyUkoly | QProblemyUkoly | QProblemyUkoly |
| Souhrn: | Seskupit | Seskupit | Seskupit | Seskupit | Sum | Sum |
| Křížová tabulka: | hlavička řádku | hlavička řádku | hlavička řádku | hlavička sloupce | hodnota | hlavička řádku |
| Řadit | | | | | | |
| Kritéria: | | | | | | |
| nebo: | | | | | | ▼ |
| | • | | | | | F |

Obr. 16. Křížový dotaz v návrhovém zobrazení

TRANSFORM Sum(QProblemyUkoly.DobaPrace) AS Hodnota SELECT QProblemyUkoly.ID, QProblemyUkoly.Osoba, QProblemyUkoly.Nazev, Sum(QProblemyUkoly.DobaPrace) AS DobaPraceCelkem FROM QProblemyUkoly GROUP BY QProblemyUkoly.ID, QProblemyUkoly.Osoba,

```
OProblemyUkoly.Nazev
```

```
PIVOT QProblemyUkoly.Resitel;
```

Na obr. 16 je zobrazen výsledný křížový dotaz v návrhovém zobrazení. Všimněte si zejména pojmenování agregovaných položek a realizace součtu hodnot v jednom řádku.

Použití položek v křížovém dotazu je určeno vlastností *Křížová tabulka* a potřebné agregační funkce vlastností *Souhrn*.

Vytvářecí dotaz QHotoveVytvorit

Vytvářecí dotaz vytvoří novou tabulku na základě provedeného výběrového dotazu a uloží do ní vyhovující záznamy. V našem případě vybereme všechny dokončené problémy a jejich údaje zapíšeme do nové tabulky *ProblemyHotove*. Navíc přidáme ještě položku *Zaloha*, obsahující datum a čas zařazení záznamu do této tabulky.

Postup práce je obdobný tvorbě výběrového dotazu. Určíme zdrojový objekt (*Problemy*), a v dolní části dotazu jednotlivé položky ve vytvářené tabulce, viz obr. 17. Položky, které nepožadujeme zobrazit, nebudou ve vytvořené tabulce existovat. Po určení druhu dotazu je navíc potřeba určit název vytvářené tabulky, viz obr. 18. (název bude zapsán do vlastnosti dotazu *Cílová tabulka*, a *Cílová databáze*).

```
SQL podoba dotazu je pouze rozšířením dotazu SELECT:
SELECT Problemy.*, Now() AS Zaloha INTO
ProblemyHotove
FROM Problemy
WHERE (((Problemy.Hotovo)=True));
```

Pozn. Při opakovaném spuštění dotazu bude stávající tabulka včetně jejích záznamů zrušena a nahrazena novou. Pro přidání dalších záznamů je třeba použít jiný typ dotazu (**Přidávací** dotaz)







Obr. 18. Určení názvu nové tabulky

Přidávací dotaz QHotovePresunout

Úkolem dalšího dotazu je přidat údaje o ukončených problémech do tabulky *ProblemyHotove*. Ale pozor, je třeba zajistit, aby do tabulky nebyly přenášeny záznamy o problémech, které tam již jsou!

K vyřešení problému použijeme přidávací dotaz, jehož zdrojem budou tabulky *Problemy* a *ProblemyHotove*, spojené přes rovnost položek *ID* levým polospojením. To je důležité, aby na výstupu byly zobrazeny i záznamy tabulky *Problemy*, které se v tabulce *ProblemyHotove* nenacházejí. Aby to byly jen ony, přidáme požadavek, aby hodnota položky *ID* z tabulky *ProblemyHotove* byla prázdná (*Is Null*). Ostatní vystupující položky jsou obdobné jako v předchozím příkladu. Navíc je třeba určit také položku cílové tabulky, do které se mají jejich hodnoty uložit (vlastnost *Přidat do*), viz obr. 19.

SQL dotaz obsahuje určení cílové tabulky a dotaz SELECT určující vkládaná data:

```
INSERT INTO ProblemyHotove ( Zaloha )
SELECT Problemy.*, Now() AS Zaloha
FROM Problemy LEFT JOIN ProblemyHotove ON
    Problemy.ID = ProblemyHotove.ID
WHERE (((Problemy.Hotovo)=True) AND
    ((ProblemyHotove.ID) Is Null));
```



Obr. 19. Přidávací dotaz

Pozn. Při práci s dotazem si všimněte funkce Návrhového zobrazení, které zobrazí záznamy vyhovující požadavkům (část dotazu za klauzulí SELECT). Skutečné přidání záznamů bude provedeno až po spuštění dotazu z Návrhového zobrazení (tlačítko Spustit z Panelu nástrojů Návrh dotazu) nebo po jeho otevření z okna databáze.
Odstraňovací dotaz QHotoveOdstranit

Záznamy zkopírované do tabulky *ProblemyHotove* je třeba z tabulky *Problemy* odstranit. To je také smysl této operace. K odstranění použijeme odstraňovací dotaz, který zruší všechny záznamy z tabulky *Problemy*, které mají hodnotu položky *Hotovo* rovnu True (K odstranění úkolů souvisejících s problémem dojde díky nastavení kaskádního odstranění vazby mezi tabulkami v okně *Relace*).

| 🛱 QHotoveOd | stranit : Odstraňovací dotaz 💶 🗖 🗙 |
|---|------------------------------------|
| Problemy Cena Termin Hotovo URL OLE | |
| Pole: Tabulka: Odstranit: Kritéria: nebo: | Hotovo Problemy kde TRUE |

Obr. 20. Odstraňovací dotaz

DELETE Problemy.Hotovo
FROM Problemy
WHERE (((Problemy.Hotovo)=True));

Aktualizační dotaz QDatumOprava

Díky ručnímu zadávání dat do tabulek se může stát, že u problému bude zapsán termín splnění dřívější, než datum ukončení některého z jeho úkolů. K odstranění tohoto problému použijeme *aktualizační dotaz*, který termín nahradí datem ukončení úkolu (správně bychom měli najít největší z datumů ukončení úkolů příslušného problému).



Obr. 21. Aktualizační dotaz UPDATE Problemy LEFT JOIN Ukoly ON Problemy.ID = Ukoly.IDProblem SET Problemy.Termin = [DatumDo] WHERE (((Problemy.Termin)<[DatumDo]));

Sjednocovací dotaz QProblemyVsechny

V některých případech budeme potřebovat zobrazení všech řešených problémů společně, tedy jak aktuálně řešených (uložených v tabulce *Problemy*), tak řešených dříve (uložených v tabulce *ProblemyHotove*).

Pozn.: Na tomto místě může být položena otázka, zda byla správně provedena analýza úlohy (zavedení tabulky **ProblemyHotove**), zda by nestačilo použít jen správné zobrazování záznamů tabulky **Problemy** (Hotovo=True, Hotovo=False a bez podmínky). Námi zvolené řešení může mít důvod v bezpečnostních požadavcích. Např. pokud ke všem úkolům má mít přístup jen omezený okruh osob, zatímco k aktuálním všichni.

K vyřešení problému použijeme sjednocovací dotaz (UNION), který dokáže do výsledné tabulky zařadit záznamy ze dvou zdrojů (tabulek nebo výběrových dotazů - SELECT). Oba zdroje se definují v rámci dotazu UNION a to je také důvodem, proč není možno tento dotaz vytvářet v grafickém prostředí QBE. Na zdroje jsou kladeny následující požadavky:

- zdroje musí mít stejný počet položek,
- názvy položek se mohou lišit, budou převzaty z prvního zdroje,
- datové typy položek se mohou lišit, zobrazené záznamy není možno editovat,
- záznamy je možno seřadit v jednotlivých zdrojích nebo všechny podle určených položek prvního zdroje.

Dotaz bude třeba zapsat v *Zobrazení SQL*:

```
SELECT ID, Osoba, Nazev, Cena, Termin FROM
Problemy
```

UNION

```
SELECT ID, Osoba, Nazev, Cena, Termin FROM
    ProblemyHotove
ORDER BY ID;
```

Definiční dotaz QHotovePrimarniKlic

Další skupinou dotazů, které není možno vytvářet v grafickém prostředí QBE jsou dotazy definiční. Můžeme je využít zejména k vytváření či úpravě struktury tabulek a jejich klíčů.

Např. pro definici primárního klíče v tabulce ProblemyHotove použijeme dotaz:

CREATE UNIQUE INDEX PrimaryKey ON ProblemyHotove (ID) WITH PRIMARY;

Pro jeho opětovné odstranění dotaz: DROP INDEX PrimaryKey ON ProblemyHotove;

Pozn.: Bližší informace o Vytvářecích dotazech, stejně jako o Předávacích dotazech je k dispozici ve velmi podrobné nápovědě k programu MS-Access.

Použití vložených dotazů (QUkolyDlouhe)

Dalším problémem je zobrazení jen těch úkolů, jejichž doba řešení přesahuje průměrnou hodnotu úkolů stejného problému. Je třeba zvládnout dva problémy:

- 1. výpočet průměrné doby řešení úkolu pro problém,
- 2. zobrazení jen těch úkolů, jejichž doba řešení přesahuje průměr.

Je zřejmé, že vypočtenou průměrnou dobu řešení úkolů pro každý problém jsme schopni získat agregačním dotazem, ale potřebujeme ji použít jako argument v klauzuli WHERE výsledného výběrového dotazu. Naštěstí je to možné, viz obr. 22 (vložený dotaz je zapsán polotučnou kurzívou):

SELECT QProblemyUkoly.*

FROM QProblemyUkoly

WHERE (((QProblemyUkoly.DobaPrace)>(SELECT

Avg(QPom.DobaPrace) AS
PrumernaDobaPrace FROM QProblemyUkoly
AS QPom WHERE
(((QPom.ID)=(QProblemyUkoly.ID)));));

Pozn.: Všimněte si přejmenování zdroje ve vloženém dotazu na QPom. To umožňuje definovat podmínku ve vloženém dotazu, odkazující se na problém, kterému právě zpracovávaný úkol patří.

| 📰 QUkolyDlou | he : Výběrový dota | z | |
|--|---|-----------------------------|-----------|
| Q Problem * ID Osoba Nazev Popis | nyLIkoly | | ▲ |
| Pole: Tabulka: Řadit: Zobrazit: Kritéria: nebo: | QProblemyUkoly.* QProblemyUkoly ✓ | DobaPrace QProblemyUkoly | Pom.D |

Obr. 22. Výběrový dotaz s vloženým dotazem

Použití parametrů (QUkolyVObdobi)

V posledním řešeném problému je třeba zobrazit jen ty úkoly, jejichž řešení spadá do určeného období. Toto období se bude zadávat jako rozmezí dvou datumů (začátku a konce). Přitom musíme vyřešit dva problémy:

- 1. zadat datum začátku a konce sledovaného období,
- 2. určit zda zpracovávaný úkol patří do sledovaného období.

Datum začátku a konce sledovaného období vložíme do dotazu jako parametry, tedy zapíšeme do hranatých závorek název parametru, viz obr. 24. Systém zjistí, že se nejedná o název

žádné z položek zdroje a proto se dotáže na hodnotu parametru, viz obr. 26. Přitom systém vypíše pouze název parametru, proto se jako název parametru často používají texty vysvětlující také jeho význam. V našem případě to budou parametry:

- [Zadejte datum začátku sledovaného období]
- [Zadejte datum konce sledovaného období]



Obr. 23. Určení příslušnosti úkolu ke sledovanému období

Pro určení, zda zpracovávaný úkol patří do sledovaného období je třeba určit, zda odpovídá některému ze stavů, zobrazených na obr. 23. Úkol může ležet uvnitř období (1), obklopovat jeden z okrajů (2 a 3) nebo současně oba okraje (4). Přitom poslední případ současně vyhovuje podmínkám (2) i (3) a nemusíme ho uvažovat.



Obr. 24. Dotaz používající parametry

Realizace kritérií pro výběr záznamů je zřejmá z obr. 24. Zbývají jen dva drobné problémy:

- 1. v jakém pořadí budou hodnoty parametrů vyžádány od uživatele,
- 2. zda bude systém kontrolovat, že jsou zadány datumy.

Oba problémy vyřešíme zařazením parametrů do seznamu *Parametrů dotazu*, viz obr. 25. Tím určíme pořadí vyžádání parametrů (shora dolů) i typovou kontrolu (nastavením datového typu parametru). Systém pak nedovolí zapsat hodnotu parametru odporující požadovanému datovému typu.



Obr. 25. Seznam parametrů dotazu

| Zadat hodnotu parametru | | | | |
|--|--|--|--|--|
| Zadejte datum začátku sledovaného období | | | | |
| 1. 6. 1999 | | | | |
| OK Storpe | | | | |
| | | | | |

Obr. 26. Určení hodnoty parametru po zobrazení dotazu

Pozn.: Z výše uvedeného vyplývá, že pokud při tvorbě dotazu nesprávně zapíšeme název položky, bude automaticky chápána jako parametr a systém se při otevření (zobrazení, spuštění) dotazu dotáže na hodnotu tohoto parametru. Výsledný SQL dotaz má tvar: PARAMETERS [Zadejte datum začátku sledovaného období] DateTime, [Zadejte datum konce sledovaného obdobíl DateTime; SELECT OProblemyUkoly.* FROM OProblemyUkoly WHERE (((QProblemyUkoly.DatumOd)>=[Zadeite datum začátku sledovaného obdobíl) AND ((QProblemyUkoly.DatumDo) <= [Zadejte datum konce sledovaného obdobíl)) OR (((OProblemyUkoly.DatumOd) <= [Zadejte datum začátku sledovaného obdobíl) AND ((QProblemyUkoly.DatumDo)>=[Zadeite datum začátku sledovaného období])) OR (((QProblemyUkoly.DatumOd) <= [Zadejte datum konce sledovaného obdobíl) AND ((QProblemyUkoly.DatumDo)>=[Zadejte datum konce sledovaného období]));

6. Formuláře (Lab03)

Formuláře slouží k zobrazení dat na obrazovce a zejména k jejich vkládání a úpravě. Za tímto účelem jsou vytvořeny různé typy objektů formuláře pro zpřístupnění položek zdrojového objektu formuláře (tabulky, výběrového dotazu, případně křížového dotazu).



Obr. 27. Souprava nástrojů

Pro podporu automatizace či odstranění opakovaných činností jsou formuláře vybaveny také možností obsloužit různé události vznikající při práci s formulářem definovanou činností (při otevření formuláře, před aktualizací záznamu, ...). K dispozici jsou tři možnosti obsluhy událostí:

- *vyhodnocení výrazu* (zejména v něm obsažených funkcí, které můžeme naprogramovat), jehož výsledek však není kam uložit,
- *spuštění makra*, tato technika se však opouští, je pozůstatkem vývojově starších verzí MS-Access,
- provedení událostní procedury, tedy programové rutiny speciálně vytvořené pro obsluhu právě určené události a jen v aktuálním formuláři. Událostní procedury jsou shromážděny ve speciálním programovém modulu, spojeném s formulářem (*Modul třídy*).

Tabelární formulář FOsoby

Pro vytvoření formuláře k editaci evidovaných osob využijeme služby průvodce formulářem, viz obr. 28. Zadáme pouze zdroj záznamů – tabulku *Osoby* a zvolíme *automatický formulář: tabelární*. Systém vytvoří celý formulář a zobrazí ho ve formulářovém zobrazení.

My přejdeme do návrhového zobrazení a provedeme několik úprav. Jednak upravíme barvy jednotlivých oblastí, vzhled vložených objektů, ale hlavně vložíme do zápatí formuláře ovládací tlačítko pro jeho uzavření.

| Nový formulář | ? × |
|--|---|
| Tento průvodce vytvoří automaticky tabelární formulář. | Návrhové zobrazení Průvodce formulářem Automatický formulář: sloupcový Automatický formulář: tabelární Automatický formulář: datový list Průvodce grafem Průvodce kontingenční tabulkou |
| Vyberte tabulku nebo dotaz, odkud pocházejí data objektu: | Osoby |
| | OK Storno |

Obr. 28. Tvorba nového formuláře

| 8 | Osoby : Formulář | _ 🗆 × |
|-------|---------------------|-------------|
| | 1 2 3 4 5 | · I · 6 · 🔺 |
| | 🗲 Záhlaví formuláře | |
| : | Seznam osob | |
| | ✓ Tělo | |
| E | Osoba | |
| | 🗲 Zápatí formuláře | |
| E | | |
| ll: I | | |
| | | |
| | | |

Obr. 29. Tabelární formulář v návrhovém zobrazení

Ke vložení tlačítka použijeme opět průvodce, který ve třech krocích vytvoří příkazové tlačítko:

- 1. volba požadované činnosti, viz obr. 30,
- 2. určení textu v tlačítku nebo zobrazení obrázku,
- pojmenování tlačítka (z názvu objektu a obsluhované události bude odvozen název procedury události, kterou průvodce vytvoří - Private Sub Konec_Click()

| Průvodce příkazovým tla | očítkem | | | | | | |
|--|--|--|--|--|--|--|--|
| Ukázka: | Co se má provést po stisku tlačítka? Pro každou kategorii činností jsou dostupné různé akce. | | | | | | |
| | Kategorie: Pohyb po záznamech Operace se záznamy Operace s formuláři Operace se sestavami Aplikace Různé | Akce: Obnovit data formuláře Otevřít formulář Použít filtr formuláře Tisknout aktuální formulář Tisknout formulář Upravit filtr formuláře Zavřít formulář | | | | | |
| Storno <⊒pět <u>D</u> alší > Do <u>k</u> ončit | | | | | | | |

Obr. 30 Průvodce příkazovým tlačítkem – krok 1

Na závěr můžeme zkontrolovat a doplnit hodnoty vlastností formuláře, viz. obr. 31 a případně vlastností jednotlivých oblastí (sekcí), viz obr. 32.

| 💣 Formulář | | | | | | × |
|-------------------------------|----------------------|---------|--------|---------------|---------|---|
| Formátové | Datové | Udál | ostní | Jiné | Všechny |) |
| Titulek | | | Osoby | | | |
| Výchozí zobr | azení | | nekon | ečný formulář | | |
| Povolená zoł | orazení | | formul | ářové | | |
| Posuvníky | | | oba | | | |
| Volič záznam | ů | | ano | | | |
| Navigační tla | ičitka | | ano | | _ | |
| Dělicí čáry | | | ano | | _ | |
| Automaticky | změnit velikos | ;t | ano | | _ | |
| Automaticky | zarovnat na s | střed . | ano | | | |
| Styl okraje . | | | promě | nný | _ | |
| Ovládaci nat | oidka | | ano | | | |
| Min. a max. I | tlačitka | | obé po | ovolena | _ | |
| Zaviraci tlači | tko | | ano | | _ | |
| – Hacitko Co je – ă∼i | 3 | • • • • | ne | | _ | |
| Sirka | | | 6,8980 | IM D | _ | |
| UDrazek | • • • • • • • • • • | | (zadny | /) | _ | |
| Typ obrazku De žise velike | | | viozen | γ | _ | |
| Rezim velikos | sti obrazku Vézek | | deeve | UC - 6% | _ | |
| Obrásku vod | razek Ilo sobo | | aopros | screa | _ | |
| Dělepí odu x | ne sebe | | 10 | | _ | |
| Delení osy x Dělepí osy v | mnzky mřížku | | 10 | | _ | |
| Bozyržepí pr | o tiek | | ne Io | | | |
| Zdroj palety | | | (výcho | ozi) | | - |

| Caption |
|--------------------|
| Default View |
| Views Allowed |
| Scroll Bars |
| Record Selectors |
| Navigation Buttons |
| Dividing Lines |
| Auto Resize |
| Auto Center |
| Border Style |
| Control Box |
| Min Max Buttons |
| Close Button |
| Whats This Button |
| Width |
| Picture |
| Picture Type |
| Picture Size Mode |
| Picture Alignment |
| Picture Tiling |
| Grid X |
| Grid Y |
| Layout For Print |
| Palette Source |
| |

| 💣 Formulář | : | | | | | | X |
|-----------------------|--------|------|---------|----------|---|----------|---|
| Formátové | Datové | Udál | ostní 🏻 | Jiné | 1 | Všechny | 1 |
| Zdroj záznar Filtr | nů | | Osoby | | | T | |
| Řadit podle | | | Osoba | | | | |
| Povolit filtry | | | ano | | | | |
| Povolit úpra | vy | | ano | | | | |
| Povolit odstr | ranění | | ano | | | | |
| Povolit přidá | vání | | ano | | | | |
| Zadat data . | | | ne | | | | |
| Typ sady zá | znamů | | dynami | cká sada | | | |
| Uzamčení zá | znamů | | žádné u | uzamčení | | | |

Layout For Print Palette Source Filter Order By Allow Filters Allow Edits Allow Deletions Allow Additions Data Entry Recordset Type Record Locks

| 🖥 Formulář | | × | |
|---------------------------------|------------|-----------|--------------------|
| Formátové Datové Udák | ostní Jiné | Všechny | |
| Při události Current | | · · · · · | Current |
| Před vložením | | | Before Insert |
| Po vložení | | | After Insert |
| Před aktualizací | | | Before Update |
| Po aktualizaci | | | After Update |
| Při odstranění | | | Delete |
| Před potvrzením odstranění | | | Before Del Confirm |
| Po potvrzení odstranění | | | After Del Confirm |
| Při otevření | | | Open |
| Při zavedení | | | Load |
| Při změně velikosti | | | Resize |
| Při vyřazení | | | Unload |
| Při zavření | | | Close |
| Při aktivaci | | | Activate |
| Při deaktivaci | | | Deactivate |
| Při získání fokusu | | | Got Focus |
| Při ztrátě fokusu | | | Lost Focus |
| Při klepnutí | | | Click |
| Při poklepnutí | | | Dbl Click |
| Při stisku tlačítka myši | | | Mouse Down |
| Při přesunu myši | | | Mouse Move |
| Při uvolnění tlačitka myši | | | Mouse Up |
| Při stisku klávesy | | | Key Down |
| Při uvolnění klávesy | | | Key Up |
| Při úhozu na klávesu | | | Key Press |
| Přednost klávesových událostí . | ne | | Key Preview |
| Při chybě | | | Error |
| Při filtru | | | Filter |
| Při použití filtru | | | Apply Filter |
| Při události časovače | | | Timer |
| Interval časovače | 0 | | Timer Interval |

Obr. 31a. Vlastnosti formuláře

| 💣 Formulář | | | | × | 1 |
|-------------------|----------------|------------|--------------|---------|-----|
| Formátové 🗍 🛛 | Datové | Událostn | í Jiné | Všechny | |
| Překryvné okno | | ne | | - | |
| Modální okno | | and | I | | |
| Koloběh | | vše | chny záznamy | | Ľ |
| Řádek nabídek . | | | | | Ľ |
| Panel nástrojů . | | | | | Ľ |
| Místní nabídka . | | and | I | | II. |
| Řádek místní nal | bídky | | | | II. |
| Rychlý tisk na la | iserové tisł | kárně, and | I | | II: |
| Soubor nápověc | ј у | | | | II: |
| ID kontextové n | nápovědy . | 0 | | | |
| Identifikace | | | | | ľ |
| Obsahuje modul | | and | | | |
| | | | | | |

Pop Up Modal Cycle Menu Bar Foolbar Shortcut Menu Shortcut Menu Bar Fast Laser Printing Help File Help Context ID Fag Has Module

Obr. 31b. Vlastnosti formuláře

| 💣 Sekce: Tělo | | | × | |
|---|--|---------|---|--|
| Formátové Datové | Událostní 🗍 Jiné | Všechny | | |
| Název Odstránkovat Nový řádek nebo sloupec Udržovat pohromadě Zobrazit Podmínky zobrazení | Tělo ne ne ne ano vždy | | Nai For Ne Kee Vis Dis | me ce New Page w Row Or Co ep Together sible splay When |
| Možnost zvětšení Možnost zmenšení Výška Barva pozadí Zvláštní efekt Identifikace Při klepnutí Při poklepnutí Při stisku tlačitka myši Při přesunu myši | ne ne 0,651cm -2147483633 plochý | | Can Can Hei Bao Spo Tag Cli Db Mo Mo | 1 Grow 1 Shrink ight ck Color ccial Effect g ck l Click puse Down puse Move |

Obr. 32. Vlastnosti sekce formuláře



Obr. 33. Formulář ve formulářovém zobrazení

Datový list FUkolySub

Pro práci s úkoly potřebujeme zajistit zobrazení jak jednoho úkolu, tak jeho podřízených úkolů. Jinak by ani nebylo možno podřízené úkoly vkládat. Práci začneme vytvořením formuláře pro podřízené úkoly. Zvolíme zdrojový objekt – tabulku *Ukoly* a v průvodci *automatický formulář: datový list*. Výsledkem je formulář obdobného vzhledu jako tabulka (obr. 34), funkcionalita formuláře mu však zůstala, jak záhy uvidíme.

Všechny potřebné úpravy provedeme v zobrazení datového listu, službami v nabídce *Formát*. Především je to skrytí položek (sloupců) *ID*, *IDProblem* a *IDMaster*, které uživateli stejně nic neříkají. Pokud bychom se k nim chtěli vrátit, využijeme nabídku *Zobrazit skryté sloupce*, viz obr. 35.

Dále upravíme *Písmo* použité při zobrazení formuláře. Opět tak můžeme učinit pouze na tomto místě. Případně upravíme zobrazení *Buňky*, především jejich rámování, *Výšku řádku* nebo *Šířku sloupce* v návaznosti na použitou velikost písma.

V návrhovém zobrazení potřebujeme speciální úpravu, je třeba změnit zdroj záznamů formuláře, aby zobrazoval jen podřízené úkoly a nikoliv úkoly hlavní. Protože jsme si nepřipravili potřebný dotaz jako objekt, zapíšeme příslušný SQL dotaz přímo do vlastnosti *Zdroj záznamů*:

```
SELECT Ukoly.* FROM Ukoly WHERE
  (((Ukoly.IDMaster)<>[ID]));
```

| | Ukoly | | | |
|---------------|----------------|-------------------|-------------------|---|
| | Osoba | DatumOd | DatumDo | Reseni |
| | Kotrba, Matěj | 23.3.1999 | 24.3.1999 | Datová analýza |
| | Kotrba, Matěj | 23.3.1999 | 25.3.1999 | Analýza objektů |
| | Kotrba, Matěj | 24.3.1999 | 27.3.1999 | Analýza vazeb mezi objekty |
| | Kotrba, Matěj | 27.3.1999 | 30.3.1999 | Optimalizace rozvržení dat |
| | Kotrba, Matěj | 30.3.1999 | 2.4.1999 | Návrh tabulek, definice položek |
| | Kotrba, Matěj | 3.4.1999 | 5.4.1999 | Definice validačních podmínek |
| Kotrba, Matěj | | 5.4.1999 | 6.4.1999 | Tvorba vazeb mezi tabulkami, referenční |
| | Babík, Jan | 15.4.1999 | 16.4.1999 | Propojení objektů, tvorba obsluhy |
| | Babík, Jan | 15.4.1999 | 18.4.1999 | Realizace řádků nabídek |
| | Babík, Jan | 15.4.1999 | 19.4.1999 | Realizace místních nabídek |
| | Babík, Jan | 15.4.1999 | 19.4.1999 | Obsluha horkých kláves |
| | Bublík, Václav | 15.4.1999 | 20.4.1999 | Definice činnosti po sputění úlohy |
| | Bublík, Václav | 23.3.1999 | | Nalezení kupujícího |
| | Babík, Jan | 6.4.1999 14:00:00 | 6.4.1999 16:30:00 | Kontrola integritních omezení |
| * | | 7.9.1999 | | |
| Zá | znam: 🚺 🔳 | 1 | * z 14 | |

| Obr. 34. Formulář v zobrazení datového list |
|---|
|---|



Sloupcový formulář FUkoly

Jako hlavní formulář pro evidenci úkolů vytvoříme nový formulář se zdrojem *Ukoly* typu *automatický formulář: sloupcový*. Vytvořený formulář v návrhovém zobrazení opět upravíme, zejména rozmístění objektů. S tím souvisí také pořadí, v jakém prochází objekty kurzor. Můžeme ho změnit v okně *Pořadí prvků*, viz obr. 36, vždy pro zvolenou sekci formuláře.



Obr. 36. Určení pořadí objektů

Ve formuláři opět skryjeme objekty *ID*, *IDProblem* a *IDMaster*, ale tentokrát v návrhovém zobrazení, nastavením

vlastnosti *Zobrazit*. Dále přidáme nový objekt Textové pole s názvem *TypUkolu*, jehož zdrojem bude výraz:

```
=IIf(IsNull([ID]);"Nový";IIf([ID]=[IDMaste
r];"Hlavní";"Podřízený"))
```

Tím bude uživatel informován o tom, zda aktuální úkol je hlavní nebo již jinému úkolu podřízený.

Dále vložíme do formuláře již připravený podformulář *FUkolySub*. Průvodce nám nabídne existující formuláře a v druhém kroku způsob propojení záznamů formuláře a podformuláře. Pro nás platí "*Zobrazit objekt <příkaz SQL> pro záznamy z objektu Ukoly pomocí pole ID*". To znamená, že z aktuálního záznamu formuláře bude převzata hodnota položky *ID* a v podformuláři budou zobrazeny jen ty záznamy, jejichž položka *IDMaster* se s touto hodnotou shoduje. Totéž je možno nastavit také ručně, pokud v průvodci zvolíme *Definovat vlastní*. Současně se tato hodnota vloží automaticky do položky *IDMaster* nově vytvořeného záznamu v podformuláři. Jinak by podúkol nebylo možno zapsat (porušení kontroly referenční integrity).

Bohužel zatím není možno zadat nový záznam (hlavní úkol) do vytvořeného formuláře. Brání tomu stejná kontrola referenční integrity, tedy nevyplnění položky *IDMaster* v hlavním formuláři. Tady musíme zvolit jiný postup a vytvořit proceduru události formuláře *Před aktualizací*, která provede příkaz:

If Me.IDMaster < 1 Then Me.IDMaster = Me.ID

Výsledný formulář je na obr. 37.

| 88 | FUkoly : Formulář | | 3 |
|--------------|---------------------|--|---|
| | + + + 1 + + + 2 + + | • 3 • 1 • 4 • 1 • 5 • 1 • 6 • 1 • 7 • 1 • 8 • 1 • 9 • 1 • 10 • 1 • 11 • 1 • 12 • 1 • 13 • 1 • 14 🗖 | |
| | 🗲 Záhlaví formuláře | | 1 |
| | 🗲 Tělo | | |
| E | Odpovědná osoba | Osoba 🗾 D IDProblem IDMaster ";Iff([ID]=[IDMaster];") | |
| | Začátek řešení | DatumOd Konec řešení DatumDo | |
| - | Popis řešení úkolu | Reseni | |
| 2 | Podřízehé úkoly: | | |
| : | FUkolySub | | |
| ³ | | Dceřiné propojovací pole: IDMaster | |
| . | | Řídicí propojovací pole: ID | |
| ÷ | | | |
| 5 | | | |
| - | | | |
| | 🗲 Zápatí formuláře | • | 1 |
| | | | |

Obr. 37. Původně sloupcový formulář po úpravách

Pozn.: Při pokusu o vložení nového úkolu, ať do hlavního formuláře nebo do podformuláře bude oznámeno porušení referenční integrity. Tentokrát je na vině nevyplnění položky **IDProblem**. Navíc tuto hodnotu není odkud získat. Proto problém vyřešíme až po vytvoření následujícího formuláře pro editaci údajů problémů.

Formulář FProblemy

Formulář pro zobrazení problémů bude současně hlavním formulářem celé úlohy. Jako podformulář do něj vložíme již připravený formulář *FUkoly* (včetně do něj vloženého podformuláře *FUkolySub*). Pro ostatní vytvořené formuláře a později výstupní sestavy sem přidáme příkazová tlačítka, která je budou otevírat.

Jako obvykle použijeme dostupnou podporu, tentokrát zadáme zdrojový objekt *Problemy* a zvolíme *průvodce formulářem*. Ten nás provede jednotlivými kroky:

1. určíme zobrazené položky, v našem případě všechny položky tabulky *Problemy*,

2. zvolíme rozvržení formuláře *zarovnané* (tato volba jako jediná není přístupná přímo jako automatický formulář),

3. vybereme grafický vzhled formuláře, např. *Standardní* (ten budou nadále používat také automatické formuláře),

4. Naposledy určíme název formuláře.

Jako obvykle upravíme rozmístění položek, zakážeme zobrazení objektu *ID*. Pro větší výraznost změníme typ objektu pro položku Hotovo ze *Zaškrtávacího políčka* na *Přepínací tlačítko* s textem "*Problém ukončen*".

Do formuláře vložíme podformulář FUkoly s propojením "*Zobrazit objekt Ukoly pro záznamy z objektu Problemy pomocí pole ID*". Položka *IDProblem* podformuláře bude

spojena s položkou *ID* hlavního formuláře. Od této chvíle již můžeme vytvořit nový záznam hlavního úkolu u problému.

Bohužel podřízený úkol ještě vytvořit nelze. K dořešení tohoto problému se musíme vrátit k formuláři *FUkolySub* a vytvořit událostní proceduru pro událost *Před aktualizací*:

Private Sub Form_BeforeUpdate(Cancel As Integer)

```
If Me.IDProblem < 1 Then Me.IDProblem =
Forms!FProblemy!ID</pre>
```

End Sub

Pozn.: Tento problém je také možno vyřešit na úrovni formuláře **FUkoly**. Při vložení podformuláře **FUkolySub** bychom zadali spojení ručně pomocí dvou položek (**ID** a **IDProblem** u hlavního formuláře, **IDMaster** a **IDProblem** u podformuláře).

Do záhlaví formuláře umístíme jednak popisek "*Přehled řešení problémů*" a hlavně Pole se seznamem pro rychlé vyhledávání problémů. Se sestavením nám opět pomůže průvodce:

- 1. Zvolíme, že hodnoty pro pole budou načteny z polí na formuláři a podle výběru se nastaví aktuální záznam formuláře.
- 2. Zobrazíme položky *Nazev* a *Termin*.
- 3. Nastavíme požadované šířky sloupců, klíčový sloupec (*ID*, bude vložen systémem pro realizaci hledání zvoleného problému) skryjeme.
- 4. Jako titulek pro pole zadáme "Přejdi na problém:".

Systém vytvoří nejen samotný objekt, ale vygeneruje také proceduru události *Po aktualizaci*:

Sub PoleSeSeznamem26_AfterUpdate()

- ' Najít záznam odpovídající ovládacímu prvku.
- Me.RecordsetClone.FindFirst "[ID] = " & Me![PoleSeSeznamem26]

Me.Bookmark =

Me.RecordsetClone.Bookmark

End Sub

Protože pole se seznamem je velmi často používaný objekt, jsou na obr. 39 uvedeny všechny jeho vlastnosti, včetně jejich hodnot.

Do zápatí formuláře umístíme již známé příkazové tlačítko pro uzavření formuláře a tlačítka pro otevírání dalších formulářů (*FOsoby* a později vytvořené formuláře). S jejich vytvořením nám pomůže průvodce. Až se seznámíme se základy programování v jazyce Visual Basic for Application, přidáme také příkazové tlačítko pro přesun ukončených problémů do záložní tabulky *ProblemyHotove*. Na obr. 38 je zobrazen formulář *FProblemy* již vybavený těmito příkazovými tlačítky.

| | FProblemy : Formulář | ٢ |
|-----------------|---|---|
| | | - |
| | 🗲 Záhlaví formuláře | |
| - - | Přehled řešení problémů Přejdi na problém Nevázaný | |
| | ✓ Tělo | |
| : | Odpovědná osoba Název problému Nazev | |
| i | | |
| H÷. | | |
| 2 | | |
| ÷ | Í Í | |
| 3 | | |
| ÷ | Cena Termin Vyresetii probleniu Okonce ii reseti probleniu Cena Termin Vyresetii probleniu Cena | |
| 11 ⁴ | WWW stránky ilešení problému | |
| | URL | |
| - - | | |
| 6 | FUkoly | |
| 7. | | |
| 8 | | |
| : | Ď die (menoie se (nales – D | |
| 9 | Ridici propojovaci pole: ID | |
| II. | | |
| 10 | | |
| - 11 | | |
| <u> "-</u> | | |
| 12 | | |
| μ | 🗲 Zápatí formuláře | |
| ÷ | Qsoby Ukončené problémy do zálohy | |
| Ŀ | | 4 |
| | | |

Obr. 38. Hlavní formulář systému (FProblemy)

| Pole se seznamem: PoleSeSeznamem26 | | | | | | | | | | |
|------------------------------------|-------------------------|-----------|---------|-------|----------|---------|------|--|--|--|
| Formátové | Datové | Udál | ostní | 1 | Jiné | Všechny | | | | |
| Formát | | | | | | | · | | | |
| Počet deseti | nných míst | | autom | atick | Ý | | | | | |
| Počet sloupo | ů | | 3 | | | | | | | |
| Hlavičky slou | pců | | ne | | | | | | | |
| Šířky sloupců | ì | | 0cm;5 | ,371 | cm;1,773 | cm | | | | |
| Počet řádků | seznamu | | 8 | | | | | | | |
| Šířka seznam | ш | | 7,143 | cm | | | | | | |
| Zobrazit | | | ano | | | | | | | |
| Podmínky zol | brazení | | vždy | | | | | | | |
| Vlevo | | | 9,799 | cm | | | | | | |
| Nahoře | | | 0,099 | cm | | | | | | |
| Šířka | | | 5,196 | cm | | | - | | | |
| Výška | | | 0,423 | cm | | | - | | | |
| Styl pozadí . | | | normá | ilní | | | - 11 | | | |
| Barva pozad | Í | | -2147 | 4836 | 543 | | - 11 | | | |
| Zvláštní efek | t | | vmáčk | snutý | , | | - | | | |
| Styl okraje . | | | plná č | ára | | | - | | | |
| Barva okraje | ••••• | | 0 | | | | - | | | |
| Siřka okraje | · · · · · · · · · · · · | | vlasov | /á čá | ra | | - 11 | | | |
| Barva popře | dí | | -2147 | 4836 | 540 | | - | | | |
| Název pisma | • • • • • • • • • • | | Arial C | E | | | - 11 | | | |
| Velikost pism | a | • • • • • | 8 | | | | - | | | |
| Houstka pisn | na | | normá | ini | | | - | | | |
| Kurziva | | • • • • | ne | | | | - | | | |
| Podtrzeni pis | ma | | ne | , | | | - | | | |
| Zarovnání te | xtu | | obecn | e | | | | | | |

Format Decimal Places Column Count Column Heads Column Widths List Rows List Width Visible Display When Left Тор Width Height Back Style Back Color Special Effect Border Style Border Color Border Width Fore Color Font Name Font Size Font Weight Font Italic Font Underline Text Align

Pole se seznamem: PoleSeSeznamem26 Formátové Datové Událostní Jiné Všechny Název PoleSeSeznamem26

| Název | PoleSeSeznamem26 |
|----------------------------|------------------|
| Text na stavovém řádku | |
| Povolit automatické opravy | ano |
| Přístup klávesou tabelátor | ano |
| Pořadové číslo prvku | 0 |
| Řádek místní nabídky | |
| Vysvětlivka | |
| ID kontextové nápovědy | 0 |
| Identifikace | |
| | |

Name Status Bar Text Allow Auto Correct Tab Stop Tab Index Shortcut Menu Bar Control Tip Text Help Context ID Tag

| 😭 Pole se : | seznamem: P | PoleSeSezr | amem26 | | × | |
|---------------|-------------|------------|---------------|------------|---|-----------------|
| Formátové | Datové | Událostní | Jiné | Všechny | | |
| Zdroj ovláda | cího prvku | | | • | | Control Source |
| Vstupní mas | ka | | | | | Input Mask |
| Typ zdroje ř | ádků | tabul | ka/dotaz | | | Row Source Type |
| Zdroj řádků | | SELE | CT DISTINCTRO | OW [Proble | | Row Source |
| Vázaný slou | рес | 1 | | _ | | Bound Column |
| Omezit na se | eznam | ano | | | | Limit to List |
| Automaticky | doplnit | ano | | | | Auto Expand |
| Výchozí hod | nota | | | | | Default Value |
| Ověřovací p | ravidlo | | | | | Validation Rule |
| Ověřovací te | ext | | | | | Validation Text |
| Zpřístupnit . | | ano | | | | Enabled |
| Uzamknout | | ne | | | | Locked |
| | | | | | | |

💣 Pole se seznamem: PoleSeSeznamem26 х Událostní Formátové Datové Všechny Jiné · ... Před aktualizací Before Update After Update Change Při nepřítomnosti v seznamu . . . Not In List Enter Exit Při získání fokusu Got Focus Při ztrátě fokusu Lost Focus Click Dbl Click Při stisku tlačitka myši Mouse Down Při přesunu myši Mouse Move Při uvolnění tlačitka myši Mouse Up Při stisku klávesy Key Down Při uvolnění klávesy Key Up Při úhozu na klávesu Key Press

Obr. 39. Vlastnosti pole se seznamem

Graf FProblemyDobaPrace

V systému MS-Access je možno velmi snadno vytvořit různé grafy pomocí aplikace MS-Graph. Vytvoříme koláčový graf znázorňující podíl jednotlivých problémů na celkové době řešení. Jako zdroj dat použijeme připravený dotaz *QProblemyDobaPrace*, který sumarizuje dobu práce jednotlivých problémů. K vytvoření nového formuláře použijeme *Průvodce grafem*:

- 1. Pro zobrazení zvolíme položky Nazev a DobaPrace.
- 2. Jako typ grafu zvolíme Výsečový graf, viz obr. 40.
- 3. Protože jsme zvolili jen dvě zobrazené položky a pouze jedna z nich je číselná, systém položky správně rozmístí pro nezávislou i závislou osu grafu, viz obr. 41.
- 4. Nakonec zadáme požadovaný název grafu "Přehled náročnosti problémů".

Systém vytvoří formulář obsahující jediný objekt *Rámeček nevázaného objektu* pro třídu OLE: *graf aplikace Microsoft Graph 97*. My ještě doplníme obvyklé tlačítko pro uzavírání formuláře, případně nastavíme vlastnost formuláře *Modální* na "ano", takže formulář bude možno opustit jen jeho uzavřením.

Výsledný formulář je na obr. 42. Další úpravy formuláře jsou možné jen v prostředí aplikace MS Graph, jak vidíme na obr. 43.

| Průvodce g | rafem | | | | | | | | |
|------------|-------|----------|----------|----------|---|--|--|--|--|
| hdu | hda | M | E | 4 | Jakého typu bude graf? Vyberte graf, který zobrazí údaje vybraných poli požadovaným způsobem. | | | | |
| L. | | | - | * | Výsečový graf | | | | |
| | | 2 | 2 | <u></u> | Výsečový graf zobrazuje vztah či poměr částí k celku. Obsahuje vždy jen jednu sěli údajů, což jej činí vhodným pro zdůraznění význačných | | | | |
| • | - | | | ٥ | momentă. | | | | |
| | | | S | Romo | < Zpět Datiĭ> Dokončit | | | | |
| | | 01 40 | | - | | | | | |







Obr. 42. Formulář s vloženým grafem

Pozn.: Pokud se rozhodnete změnit velikost objektu s grafem ve formuláři, je třeba změnit velikost grafu také v aplikaci MS-Graph, jinak se ve skutečnosti změna velikosti neprojeví.

Pozn.: Průvodce grafem dovolí vložit do grafu nejvýše šest polí. Pokud jich chcete zobrazit více, změňte SQL dotaz ve vlastnosti zdroj řádků objektu s grafem.





Kontingenční tabulka FProblemyOsoba

Jak již bylo zmíněno v kapitole věnované dotazům, místo křížového dotazu, jehož výsledkem je tabulka, je pro zpřístupnění uživatelům vhodnější formulář obsahující *kontingenční tabulku*, protože má k dispozici také ostatní nástroje formulářů.

Nyní vytvoříme takový formulář zobrazující celkovou dobu řešení úkolů pro jednotlivé problémy (po řádcích), rozdělenou podle jednotlivých řešitelů (jejich údaje budou zařazeny do sloupců), což je adekvátní dotazu *QProblemyOsoba*. Zdrojem bude dotaz *QProblemyUkoly* a pro vytvoření použijeme *Průvodce kontingenční tabulkou*.

- 1. Systém nás seznámí s postupem tvorby kontingenční tabulky.
- Vybereme všechny položky potřebné pro její vytvoření Hlavičky řádků (ID, Osoba, Nazev), Hlavičky sloupců (Resitel) i pro vyplnění vzniklé tabulky (DobaPrace).
- 3. V dalším kroku se již přesuneme do prostředí MS-Excel a rozmístíme položky na správná místa, viz obr. 44.
- 4. Poslední krok je pouze informace o dokončení práce.

Systém vytvoří formulář a vloží objekt *Rámeček nevázaného objektu* pro třídu OLE: *Microsoft Excel 97* a vloží také tlačítko pro úpravu tabulky, které se provádí v prostředí MS-Excel.



Obr. 44. Průvodce kontingenční tabulkou (MS-Excel)

Opět doplníme obvyklé tlačítko pro uzavírání formuláře, a nastavíme vlastnost formuláře *Modální* na "ano", takže formulář bude možno opustit jen jeho uzavřením.

Práci s formuláři ukončíme doplněním tlačítek pro otevírání grafu a kontingenční tabulky do formuláře *FProblemy*.

7. Sestavy (Lab04)

Výstupní sestavy (tiskové sestavy), slouží ke zpracování dat a jejich následnému vytištění na tiskárně. I když využívají řadu postupů shodných s formuláři, mají také speciální nástroje. K nim patří zejména seskupování záznamů do skupin (jako v dotazech) a vyhodnocování agregovaných funkcí položek takto seskupených záznamů.

Sestava RProblemyUkoly

První sestava bude přehledným výstupem všech dat v databázi, jejím zdrojem bude dotaz *QProblemyUkoly*. Sestava bude obsahovat vždy všechny údaje o problému (vynecháme OLE objekt) a pod nimi postupně jednotlivé úkoly seřazené podle časového sledu plnění. Dobu řešení všech úkolů sečteme a navíc požadujeme výpočet procentního podílu jednotlivých problémů na součtu všech problémů.

I když snad zadání vypadá komplikovaně, uvidíme, že *Průvodce sestavou* (obr. 45) se s ním hravě vypořádá:

- 1. vybereme všechny položky, které chceme zobrazit nebo použít pro seskupování (*ID*), kromě OLE objektu,
- průvodce nabídne seskupení záznamů podle jedné ze dvou tabulek, spojených zdrojovým dotazem, my vybereme seskupení podle tabulky *Problemy*, viz obr. 46,
- 3. další úrovně seskupování odmítneme,
- 4. ale vyžádáme si řazení záznamů ve skupině podle položek *DatumOd* a *DatumDo* (viz obr. 47); navíc odskočíme do

nabídky Souhrnné volby a vyžádáme si výpočet součtu pro položku *DobaPrace*, kromě toho budeme také žádat výpočet procent z celku pro součty a zobrazení podrobně a přehledně, viz obr. 48,

- 5. zvolíme požadovanou úpravu rozmístění objektů a orientaci sestavy na výšku (budeme také žádat zarovnání všech objektů na jednu stránku), viz obr. 49,
- 6. vybereme vhodný styl zápisu objektů sestavy,
- a nakonec uvedeme název sestavy "Řešené problémy", který bude uveden v záhlaví a použit také jako název nově vytvořené sestavy. Ten posléze upravíme podle svých požadavků.



Obr. 45. Možnosti tvorby nové sestavy

| Průvodce sestavou | |
|--|--|
| Jak choste prohižet data? sodia Problemy podie Ukoly Zobrazit dalši informace | ID, Osoba, Nazev, Popis, Cena, Temin, <u>Hotovo, URL, OLE</u> Restel, DatumOd, DatumDo, Reseni, DobaPhace |
| | omo <zpět qalii=""> Dokončit</zpět> |
| Obr. 46. Prù | vodce sestavou – krok 2 |
| Průvodce sestavou Jek choste řadit a sumarizoval inform | nace v podrobných záznamech? Záznany můžete řadit vzestupně nebo sestupně. naximálně podle čtyř poli. 1 DetumOd * 124 2 DetumDo * 124 3 * 124 4 * 125 5 ouherné vgiby |
| St | tomo <zpêt qalií=""> Dokončit</zpêt> |

Obr. 47. Průvodce sestavou – krok 4

| Pole | Sum Avg Min Max | Stomo |
|-----------|-----------------|--|
| Dobarrace | | Zobrazit Pogrobně a přehledně Jen přehledně Výpočet procent z celku pro součty |

Obr. 48. Průvodce sestavou – krok 4a

| | | <i></i> | | | Rozvizeni sestavy | Unencace |
|------------|-------------|---------|--------------------|---------|-------------------------|------------|
| ~~ / | (AAA) | n. | | | • Udsazované | • Na yyak |
| KXX | XXXX | NNNX | NNER | XXXX | C Blokové | C Na tifks |
| XIOR | XXXXXX | | | | C Hierarchické 1 | |
| | | NUMBER | RODORE | NUMBER | | |
| | | RXXXXX | RODUCE | 330333 | C Hiejarchické 2 | |
| | NECES | ***** | ***** | | C Zarovnané vlevo 1 | |
| | | XXXXXX | KODECE | XXXXX | | |
| | NECES | | | | C Zarovnané vjevo 2 | |
| | | RXXXX | RODICE | REACTED | | |
| | | 300303 | RODOLCH RODOLCH | 30000 | | |
| | | | | | | |
| | | | | | Zarovnat šířky poli tak | , aby se |
| | | | | | vsechna pole vešla na | stranku. |
| | | | | | | |
| | | | | | | |

Obr. 49. Průvodce sestavou – krok 5

Průvodcem připravená sestava sice splňuje technické požadavky, ale rozmístění objektů bude třeba upravit. Hlavně proto, že průvodce ve skutečnosti neupraví šířky objektů tak, aby se skutečně vešly na jednu stranu, viz obr. 50.

Ve speciálním okně pro seskupování a řazení záznamů (viz obr. 51) zkontrolujeme, zda je správně nastavena tvorba seskupení a řazení záznamů. Jak vidíme, liší se seskupení od řazení tím, že v jeho případě požadujeme zobrazení záhlaví nebo zápatí skupiny. Pak můžeme také určit způsob seskupování v závislosti na datovém typu zdrojové položky, viz tab. 5.

Poslední důležité nastavení je způsob rozmisťování sekcí na jednotlivé stránky. Nejčastěji se záhlaví skupin spojuje s prvním tělem. Jak budou sestaveny jednotlivé stránky je patrné z obr. 52.

Pozn.: Pokud tiskárna tiskne každou druhou stránku prázdnou, nejspíš jste nenastavili pravý okraj sestavy, nebo je nesoulad s nastavením vzhledu stránky. Rozměry se totiž nezaokrouhlují na milimetry ale na tiskové body.

| ľ | Ì | Proble | emyUko | oly : Se | estava | | | | | | | | | | | | _ 🗆 | × |
|-----------------|----|--------|-----------|-----------|---------|-----------|-----------|----------|---------|---------|-----------|-----------|-----------|-----------|--------------|------------|-----------|-----|
| | 1 | 1 | · · · ; | 2 • 1 • 3 | 3 • 1 • | 4 • • • 8 | 5 • 1 • 1 | 6 • • • | 7 * 1 * | 8 • • • | 9 • • • 1 | 0 • • • 1 | 1 • • • 1 | 2 · + • 1 | 3 • • • 1 | 14 · + • 1 | 5 · · · 6 | |
| Г | ٦Ì | 🗲 Záhl | aví sest | avy | | | | | | | | | | | | | | |
| Řešené problémy | | | | | | | | | | | | | | | | | | |
| Ľ | - | | | | | | | | | | | | | | | | | |
| | | 🗲 Záhl | aví strá | nky | | | | | | | | | | | | | | İ I |
| IH | ų | 🗲 Záhl | aví ID | | | | | | | | | | | | | | | |
| ŀ | - | 0dpa | vědná | озова | e | Osobi | ¥ | | | • | Term | ín vyře | ešení p | robléi | nu [] | ermin | | |
| | - | Náze | v prob | lému | | Nazev | | | | |] | | | | | | | |
| : | | Popis | probl | ému | | Popis | | | - | | | | | | | | | |
| | - | Cena | řešen | í prob | lému | Cena | | | | |] | | | | | | | |
| | - | | Začát | ek | | Konec | | Ĭ | ešitel | | Popis ř | ešení i | ikolu | | | Doba | Práce | |
| l | 1 | 🗲 Tělo | | 1 | 1 | 1 | 1 | 1 | | | 1 | 1 | | 1 | 1 | 1 | | |
| ║. | : | | Datum | d | | DatumD | • | R | esitel | • F | Ceseni | | | | | Do | baPrace | |
| l | I | 🗲 Záp | atí ID | 1 | | 1 | 1 | | | | 1 | 1 | , | 1 | 1 | , | i – 1 | |
| IC. | · | ="Přeh | led prob | lému " c | & "Náze | sµ′ = ″ & | ~″&Ø\ | zzev] (| £" (" & | Gunt(*, |)&""& | Цf(Cou | t(*)=l; | podrob | ný zázna | m";"poo | robné z | |
| Ш, | 1 | Celk | ová de | oba ře | šení p | roblé | mu ve | dne | ch | | | | | | um([1 | obaP. | race]) | |
| Ľ | : | Podíl | v % | | | | | | | | | | | | ονý s | pučet | Sum]) | |
| lL | | 🗲 Záp | atí strán | iky | | | 1 | | | | | | | | 1 | | | |
| | | | | | | | | | | | | | | | | | | |
| II. | 1 | =Now | p | | | | | | | | | = | Strana | " & [Pi | ge] & | " z " & į | Pages) | |
| ╟ | | | atí sesta | avy 1 | | | | | - | 1 | | 1 | 1 | 1 | | 10.2.2 | ر م | |
| 1 | - | Celko | wa do | pa reš | eni vš | ech pr | oblém | <i>u</i> | | | | | | | Sum(| 1100ai | race]) | - |
| • | | | | | | | | | | | | | | | | | ► | // |

Obr. 50. Sestava s agregovanými výpočty

| 🕼 Řazení a seskup | ování | | | × | |
|--------------------|----------|----------------|------------------------------|----------|-------------|
| Pole/Výraz | | | Pořadí řazení | | |
| | • | vzestupně | | | |
| DatumOd | | vzestupně | | | |
| DatumDo | | vzestupně | | | |
| | | | | <u> </u> | |
| | | Vlastnosti ski | ipiny | | |
| Záhlaví skupiny | ano | | | | Group Head |
| Zápatí skupiny | ano | | Označte pole nebo napište | | Group Foot |
| Seskupovat | každou H | odnotu | výraz, podle kterého se bude | | Group On |
| Interval skupiny | 1 | | řadit či seskupovat. | | Group Inter |
| Udržovat pohromadě | s prvním | tělem | | | Keen Toget |

Obr. 51. Řazení a seskupování záznamů

Tab. 5. Způsoby seskupování záznamů

| Tab 6 Nastavení vlast | nosti <i>Udržovat pohromadě</i> |
|-----------------------|---------------------------------|
| | |

| Nastavení | Popis | Visual |
|----------------|--------------------------------|--------|
| | | Basic |
| Ne | (Výchozí hodnota) Tiskne | 0 |
| | skupinu bez toho, aby | |
| | záhlaví, tělo a zápatí skupiny | |
| | nutně byly na stejné stránce. | |
| Celá skupina | Tiskne záhlaví, tělo a zápatí | 1 |
| | skupiny na stejnou stránku. | |
| S prvním tělem | Tiskne záhlaví pouze tehdy, | 2 |
| | může-li na stejné stránce být | |
| | i první záznam těla skupiny. | |

| Datový typ pole | Nastavení | Seskupení záznamů | Visual Basic |
|--------------------|---------------------------------|--|--------------|
| Text | (Výchozí hodnota) Každá hodnota | Stejné hodnoty položky nebo výrazu. | 0 |
| | Počáteční písmena | Stejných prvních n písmen položky nebo výrazu. | 1 |
| Datum/čas | (Výchozí hodnota) Každá hodnota | Stejné hodnoty položky nebo výrazu. | 0 |
| | Rok | Data ve stejném kalendářním roce. | 2 |
| | Čtvrtletí | Data ve stejném kalendářním čtvrtletí. | 3 |
| | Měsíc | Data ve stejném měsíci. | 4 |
| | Týden | Data ve stejném týdnu. | 5 |
| | Den | Data ve stejném dnu. | 6 |
| | Hodina | Časy ve stejné hodině. | 7 |
| | Minuta | Časy ve stejné minutě. | 8 |
| Čítač, Měna, Číslo | (Výchozí hodnota) Každá hodnota | Stejné hodnoty položky nebo výrazu. | 0 |
| | Interval | Hodnoty uvnitř intervalu, který určíte. | 9 |

```
Na závěr zkontrolujeme hodnoty všech vlastností sestavy, jak jsou uvedeny na obr. 53 a doplníme obsluhu události Při nepřítomnosti dat. Vyžádáme si proceduru události a zapíšeme:
```

```
Private Sub Report_NoData(Cancel As
Integer)
MsgBox "Pro sestavu '" & Me.Name & "'
nejsou k dispozici žádná data.
Sestavu nelze zobrazit.",
vbCritical, "Prázdná sestava"
Cancel = True
End Sub
```

Pokud zdrojový objekt pro sestavu neobsahuje žádná data (neexistuje žádný evidovaný problém), dojde k řadě chyb při vytváření sestavy. Na rozdíl od formuláře totiž v sestavě není vyčleněn prostor pro nový záznam, jak je tomu u formulářů, které se zobrazí bez problémů i pro prázdný zdrojový objekt.

Pozn.: Podobnost sestav a formulářů je tak velká, že bylo dříve možno uložit formulář jako sestavu. Dnes je zřejmá především shodná skladba typů objektů formuláře a sestavy. Na obr. 50 vidíme použití **Pole se seznamem** pro některé položky, přesto že rozvinutí seznamu hodnot v sestavě není možné. Při zobrazení náhledu sestavy a jejím tisku se objekt bude chovat jako **Textové pole**, proto typ objektu není třeba měnit.

| Záhlaví sestavy | |
|----------------------|--|
| Záhlaví stránky | |
| Záhlaví skupiny ID=1 | |
| Detail – úkol 1 | |
| Detail – úkol 2 | |
| Zápatí skupiny ID=1 | |
| Záhlaví skupiny ID=2 | |
| Detail – úkol 3 | |
| Detail – úkol 4 | |
| Detail – úkol 4 | |

| Záhlaví stránky |
|----------------------|
| Detail – úkol 5 |
| Zápatí skupiny ID=2 |
| Záhlaví skupiny ID=6 |
| Detail – úkol 6 |
| Zápatí skupiny ID=6 |
| Zápatí sestavy |
| |
| |
| |
| Zápatí stránky 2 |

Obr. 52. Rozmístění sekcí na stránky

| 🖆 Sestava 💦 🚺 | | | | | | | × | |
|---|--|-------|---|---|---|---------|---|--|
| Formátové | Datové | Událo | ostní | Jiné | Τ | Všechny | 1 | |
| Formátové Titulek Záhlaví strán Zápatí stránk Udržovat sku Šířka Obrázek Typ obrázku Režim velikos Zarovnat obr Obrázky ved Stránky obrá Dělení osy x Dělení osy y | Datové iky upiny pohroma sti obrázku rázek izku mřížky mřížky | Udák | ostní Řešen všechr všechr po slou 15,996 (žádný vložen oříznou dopros ne všechr 10 10 | Jiné é problémy ny stránky upci Scm () ý ut střed | | Všechny | | |
| Rozvržení pro tisk ano Zdroj palety (výchozí) | | | | | | | | |

| Caption |
|-------------------|
| Page Header |
| Page Footer |
| Grp Keep Together |
| Width |
| Picture |
| Picture Type |
| Picture Size Mode |
| Picture Alignment |
| Picture Tiling |
| Picture Pages |
| Grid X |
| Grid Y |
| Layout For Print |
| Palette Source |

| 🖆 Sestava | | × | |
|------------------|------------------|--------------------|---------------|
| Formátové Datové | Událostní 🚺 Jiné | Všechny | |
| Zdroj záznamů | SELECT DISTING | TROW [QProblem 💌 📖 | Record Source |
| Filtr | | | Filter |
| Zapnout filtr | ne | | Filter On |
| Řadit podle | | | Order By |
| Zapnout řazení | ne | | Order By On |

| 🖀 Sestava | | | | × | |
|-----------------------|-----------|------------|---------|----------|-----------|
| Formátové Datové | Událostní | Jiné | Všechny | | |
| Při otevření | | | | ▼ | Open |
| Při zavření | | | | | Close |
| Při aktivaci | | | | | Activate |
| Při deaktivaci | | | | | Deactivat |
| Při nepřítomnosti dat | [událostr | ní procedu | ra] | | No Data |
| Na stránce | | | | | Page |
| Při chybě | | | | | Error |

💣 Sestava × Jiné Formátové Datové Událostní Všechny Uzamčení záznamů žádné uzamčení -Seskupování dat použít systémová nastavení Řádek nabídek Panel nástrojů Řádek místní nabídky Rychlý tisk na laserové tiskárně . ano Soubor nápovědy ID kontextové nápovědy 0 Tag

Obsahuje modul ano

Record Locks Date Grouping Menu Bar Toolbar Shortcut Menu Bar Fast Laser Printing Help File Help Context ID Tag Has Module

| Formátové Datové Událostní Jiné Všechny Název né né | 🚰 Sekce: Tělo | | × |
|--|----------------------------|-----------------|----|
| Název Tělo Odstránkovat ne Nový řádek nebo sloupec ne Udržovat pohromadě ano Zobrazit ano Možnost zvětšení ne Možnost zmenšení ne Výška 0,677cm Barva pozadí 16777215 Zvláštní efekt plochý Identifikace Při formátování Při tisku Při ústupu | Formátové Datové Událos | tní Jiné Všechr | ער |
| Odstránkovat ne Nový řádek nebo sloupec ne Udržovat pohromadě ano Zobrazit ano Možnost zvětšení ne Možnost zmenšení ne Výška 0,677cm Barva pozadí 16777215 Zvláštní efekt plochý Identifikace Při formátování Při tisku - | Název | <u>šlo</u> | |
| Nový řádek nebo sloupec ne Udržovat pohromadě ano Zobrazit | Odstránkovat ne | 9 | |
| Udržovat pohromadě ano Zobrazit ano Možnost zvětšení ne Možnost zmenšení ne Výška 0,677cm Barva pozadí 16777215 Zvláštní efekt plochý Identifikace Při formátování Při tisku - Při ústupu - | Nový řádek nebo sloupec ne | 9 | |
| Zobrazit ano Možnost zvětšení ne Možnost zmenšení ne Výška 0,677cm Barva pozadí 16777215 Zvláštní efekt plochý Identifikace Při formátování Při tisku | Udržovat pohromadě a | 10 | |
| Možnost zvětšení ne Možnost zmenšení ne Výška 0,677cm Barva pozadí 16777215 Zvláštní efekt plochý Identifikace Při formátování Při tisku | Zobrazit ar | 10 | |
| Možnost zmenšení ne Výška 0,677cm Barva pozadí 16777215 Zvláštní efekt plochý Identifikace Při formátování Při tisku Při ústupu | Možnost zvětšení ne | 8 | |
| Výška 0,677cm Barva pozadí 16777215 Zvláštní efekt plochý Identifikace Při formátování Při tisku Při tisku Při ústupu Identifikace | Možnost zmenšení ne | 9 | |
| Barva pozadí 16777215 Zvláštní efekt plochý Identifikace Při formátování Při tisku Při tisku Při ústupu Při ústupu | Výška0, | ,677cm | |
| Zvláštní efekt plochý Identifikace | Barva pozadí 10 | 5777215 | |
| Identifikace | Zvláštní efekt pl | ochý | |
| Při formátování | Identifikace | | |
| Při tisku | Při formátování | | |
| Při ústupu | Při tisku | | |
| | Při ústupu | | _ |

Obr. 53. Vlastnosti sestavy a její sekce

Sestava RProblemyDobaPrace

Druhá sestava bude zobrazovat sumarizovaná data problémů, tedy celkovou dobu řešení problému, kterou zjišťujeme pomocí dotazu *QProblemyDobaPrace*. Zato požadujeme grafické znázornění podílu úkolu na celkové době řešení všech problémů.

Využijeme nabídku *Automatická sestava: tabelární*. Systém vytvoří celou sestavu, kterou jako vždy mírně upravíme uspořádáním objektů na stránce. Zrušíme objekt pro zobrazení položky ID, která nyní není důležitá a rozšíříme zápatí sestavy o výpočet celkové doby práce na všech problémech. Přidáme objekt *Popis* "Celková doba řešení všech problémů" a *Textové pole* s názvem "DobaPraceCelkem" a zdrojem objektu: =Sum([DobaPrace])

Tím jsme si připravili podmínky pro vytvoření grafického objektu v sekci *Tělo*, který budeme upravovat podle procentního podílu problému na celkovém součtu. Vložíme objekt *Obdélník* s názvem "Podil" vhodné barvy a šířky 5,7 cm.

Událost *Při formátování* sekce *Tělo* obslouží změnu šířky objektu *Podil*. Podíl úkolu na celkovém součtu určíme pomocí

| 1 | | RProblemyDobaPrace : Sestava |
|----|---|---|
| | | ····1····2····3····4····5····6····7····8····9····10····11····12····13····14····15····1€ |
| | | ✓ Záhlaví sestavy |
| | - | Přehled řešených problémů |
| I | ! | |
| l | | 🗲 Záhlaví stránky |
| I | - | Název problému Odpovědná osoba Doba Práce Podíl % |
| l | Ì | |
| Iſ | : | Nazev Osoba DobaPrace |
| l | | 🗲 Zápatí stránky |
| Iſ | · | |
| IL | • | =Now) = Strana " & [Page] & 'z " & [Pages] |
| IL | | ✓ Zápatí sestavy |
| | - | Celková doba řešení všech problémů Sum([DobaPrace]) |
| ŀ | j | |

Obr. 54. Sestava s objektem proměnlivé šířky

hodnot objektů DobaPrace a DobaPraceCelkem, které jsou součástí aktuální sestavy (na ni je možno se odkázat Me): Me.Podil.Width = 5.7 * 567 * (Me.DobaPrace / Me.DobaPraceCelkem)

Pozn.: Rozměry objektů jsou ukládány v tiskových bodech, přitom zhruba platí 1 cm = 567 bodů. Nesoudělnost obou jednotek způsobuje nepřesnost ve vyjádření rozměrů jak již bylo zmíněno dříve.

8. Makra (Lab05)

Makra (makropříkazy) byly původně vytvořeny jako základní prostředek pro realizaci výkonných operací při práci s databází. Zejména to byly:

- obsluha událostí při práci s formuláři a sestavami (dnes realizované především pomocí procedur událostí),
- činnost po spuštění úlohy makro AutoExec (dnes otevření určeného formuláře),
- obsluha stisku horkých kláves

 makro AutoKeys, česky
 Automatické klávesy (název tohoto makra je možno určit),
- realizace řádku nabídek uživatelského menu (dnes jsou řádky nabídek vytvářeny jako samostatné objekty, stejně jako panely nástrojů).

Z uvedených určení již zůstalo jediné, které je stále vyhrazeno makrům – obsluha *horkých kláves*. Na obr. 55. je zobrazeno makro realizující obsluhu několika horkých kláves:

- CTRL+N přechod na nový záznam (jen ve formuláři),
- CTRL+M přechod na další záznam (jen ve formuláři),
- F11 zrušení obvyklé funkce této klávesy, kterou je přechod do hlavního okna úlohy (toto makro je součástí zabezpečení úlohy před neoprávněnými zásahy).





Pozn.: V nápovědě je makro pro obsluhu horkých kláves nazýváno jednou "Automatické klávesy", podruhé "AutomatickéKlávesy". Může se ale stát, že ani jeden z nich nebude fungovat. Název makra lze totiž určit následujícím příkazem, autoři některých úloh toho využívají a nastavují název podle svých požadavků:

```
Application.SetOption "Key Assignment
Macro", "AutoKeys"
```

Pojmenování jednotlivých funkčních kláves je uvedeno v tab. 8. Jak vidíme, uzavírají se do složených závorek. Běžné klávesy se uvádějí přímo.

Pro určení klávesových příkazů spojených s nějakou z funkčních kláves SHIFT, CTRL a ALT, se používají prefixy uvedené v tab. 7.

| Klávesa | Kód |
|---------|-----|
| SHIFT | + |
| CTRL | ^ |
| ALT | % |

Tab. 8. Pojmenování funkčních kláves

| Klávesa | Kód |
|-----------------|-------------------------------|
| BACKSPACE | {BACKSPACE}, {BS} nebo {BKSP} |
| BREAK | {BREAK} |
| CAPS LOCK | {CAPSLOCK} |
| DEL nebo DELETE | {DELETE} nebo {DEL} |
| ŠIPKA DOLŮ | {DOWN} |
| END | {END} |
| ENTER | {ENTER}nebo ~ |
| ESC | {ESC} |
| HELP | {HELP} |
| HOME | {HOME} |
| INS nebo INSERT | {INSERT} nebo {INS} |
| ŠIPKA VLEVO | {LEFT} |
| NUM LOCK | {NUMLOCK} |
| PAGE DOWN | {PGDN} |
| PAGE UP | {PGUP} |
| PRINT SCREEN | {PRTSC} |
| ŠIPKA VPRAVO | {RIGHT} |
| SCROLL LOCK | {SCROLLLOCK} |
| TAB | {TAB} |
| ŠIPKA NAHORU | {UP} |
| F1, F2,, F16 | {F1}, {F2},, {F16} |

Všechny akce makropříkazů jsou dostupné také v programových rutinách pomocí objektu DoCmd. Jako metodu zde uvádíme anglický název akce. Potřebné vlastnosti se uvádí jako parametry metody:

- KopírovatObjekt CopyObject
- Maximalizovat Maximize
- Minimalizovat Minimize
- NajítDalší FindNext
- NajítZáznam FindRecord
- NastavitHodnotu SetValue
- NastavitPoložkuNabídky SetMenuItem
- NastavitVarování SetWarnings
- Obnovit Restore
- OdeslatObjekt SendObject
- OdstranitObjekt DeleteObject
- OknoSeZprávou MsgBox
- OtevřítDotaz OpenQuery
- OtevřítFormulář OpenForm
- OtevřítModul OpenModule
- OtevřítSestavu OpenReport
- OtevřítTabulku OpenTable
- PoužítFiltr ApplyFilter
- PředatKlávesovéÚhozy SendKeys
- PřejítNaOvládacíPrvek GotoControl
- PřejítNaStránku GotoPage
- PřejítNaZáznam GotoRecord
- Přejmenovat Rename
- PřekreslitObjekt RepaintObject

- PřesunVelikost MoveSize
- PřesýpacíHodiny HourGlass
- PřevéstDatabázi TransferDatabase
- PřevéstListTabulky TransferSpreadsheet
- PřevéstText TransferText
- PřidatNabídku AppendItem
- SpustitAplikaci -RunApp
- SpustitKód RunCode
- SpustitMakro -RunMacro
- SpustitPříkaz RunCommand
- SpustitSQL RunSQL
- Uložit Save
- VybratObjekt SelectObject
- Výstup OutputTo
- Vytisknout PrintOut
- ZastavitMakro StopMacro
- ZastavitVšechnaMakra StopAllMacros
- Zavřít Close
- ZavřítDatabázi Quit
- ZnovuSpustitDotaz Requery
- ZobrazitNástroje ShowToolbar
- ZobrazitVšechnyZáznamy ShowAllRecords
- ZobrazovatVýsledky Echo
- ZrušitUdálost CancelEvent
- ZvukovýSignál Beep

Řádek nabídek MenuOsoby

Pro formulář FOsoby vytvoříme speciální řádek nabídek a připojíme ho ke stejnojmenné formuláře vlastnosti Dříve bychom skladbu řádku nabídek vytvářeli pomocí maker, dnes již jako samostatný objekt. Zobrazíme nabídku **Zobrazit-**Panely nástrojů-Vlastní a vytvoříme nový panel nástrojů.

Ten přejmenujeme podle našich požadavků a určíme jeho vlastnosti. Zejména je třeba určit, že se jedná o řádek nabídek a nikoliv panel nástrojů.

Panel nástrojů postupně naplníme jednotlivými nabídkami. Buď z karty **Příkazy** nebo přetažením myší z jiného řádku nabídek (při držení klávesy CTRL, aby se jednalo o kopírování a nikoliv přesouvání!). Současně určíme vlastnosti nabídky včetně případné ikony, viz obr. 56.



Obr. 56. Tvorba řádku nabídek

| ۷ | lastnosti panelu | nástrojů | | | ? × |
|---|-----------------------------|---------------------|------|--------------------------------------|-----|
| | Vybr <u>a</u> ný panel ná: | strojů: Menu | iOso | ъру | • |
| | -Vlastnosti panelu | nástrojů | | | |
| | Náz <u>e</u> v panelu: | MenuOsoby | _ | | |
| | <u>T</u> yp: | Řádek nabío | lek | | • |
| | Stabilní u <u>m</u> ístění: | Povolit vše | _ | | • |
| | 🔽 Zobrazit v <u>n</u> a | bídce panelů | ☑ | Povolit pře <u>s</u> unutí | |
| | 🔽 Povolit úprav | У | ☑ | Povolit zobrazení či s <u>k</u> ryti | í |
| | 🔽 Povolit změnu | u <u>v</u> elikosti | | | |
| | Obnovit předv | olené | | Zavřít | |

Obr. 57. Vlastnosti řádku nabídek

Pozn.: V názvu nabídky se používá také znak &. Určuje podtržení následujícího znaku názvu a současně vyvolávání nabídky klávesovou zkratkou Alt+Znak.

Pokud nám některá činnost v nabídce (viz obr. 58) chybí, nezbývá, než vytvořit makro s potřebnou činností, viz obr. 59 a vložit do řádku nabídek z okna příkazy (obr. 58), kde je k dispozici kategorie *Všechna makra*.

Pozn.: Oddělovače skupin nabídek se zadávají jako vlastnost **Začátek skupiny** následující nabídky (viz obr. 56).

| Kat <u>eg</u> orie: Soubor Úpravy Zobrazit Datový list Záznamy Okno a nápověda Návrh tabulky Návrh dotazu Návrh formuláře či sest Vybraný příkaz: Popis | Přík <u>a</u> zy: Vlastní Nová databáze Přík <u>a</u> zy: Nová databáze Přík <u>a</u> zy: Nový objekt Nový objekt Nový objekt Nový objekt Nový objekt Upravit výběr ▼ | |
|--|--|--|
|--|--|--|

Obr. 58. Nabídka příkazů pro řádek nabídek

| 7 | 🌋 MProMenu : Makro 📃 🗖 🗙 | | | | | | |
|---------|--------------------------|-------------------|--------------------|------------|--|--|--|
| | Název makra | Podmínka | Akce | Komentář 🔺 | | | |
| | maximize | | Maximalizovat | | | | |
| | minimize | | Minimalizovat | | | | |
| restore | | | Obnovit | | | | |
| | | | OknoSeZprávou | | | | |
| | | | | | | | |
| | | | | | | | |
| | Argumenty akce | | | | | | |
| Zp | oráva vukový signál | Velika | ost okna upravena. | _ | | | |
| TN | /n | informační zpráva | | | | | |
| Til | tulek | Restore | | | | | |

Obr. 59. Makro s činnostmi pro řádek nabídek

Popsaným popisem vytvoříme celou skladbu řádku nabídek MenuOsoby:

```
Soubor
   Zavřít
Záznamv
   Filtr
       (celá obvyklá nabídka)
   Seřadit
       (celá obvyklá nabídka)
   Další
   Nový
Okno
   Nad sebe vodorovně
   Vedle sebe svisle
   Na sebe
   Uspořádat ikony
   Zvětšit (makro MProMenu.maximize)
   Obnovit (makro MProMenu.restore)
   Zmenšit (makro MProMenu.minimize)
Nápověda
   (celá obvyklá nabídka)
```

Pozn.: Aby nabídka Záznamy-Filtr pracovala stejně jak jsme u formulářů zvyklí, je třeba z kategorie Vestavěné nabídky přenést položku Záznamy do našeho řádku nabídek a teprve následně upravit její složení. Stejně přesuneme do naší nabídky položku Nápověda.

Řádek nabídek MenuSestavy

Obdobně vvtvoříme řádek nabídek pro tiskové sestavy: Soubor Zavřít Vzhled stránky Náhled Tisk Okno Nad sebe vodorovně Vedle sebe svisle Na sebe Uspořádat ikony Zvětšit (makro MProMenu.maximize) Obnovit (makro MProMenu.restore) Zmenšit (makro MProMenu.minimize) Nápověda (celá obvyklá nabídka)

Pozn.: Položky Okno a Nápověda mají stejné složení jako v řádku nabídek MenuOsoby. Můžeme je tedy z tohoto řádku přenést myší do nové nabídky. Nezapomeňte na držení klávesy CTRL, aby se jednalo o kopírování a ne přesun.

Otevření požadovaného řádku nabídek spolu s formulářem nebo sestavou zajistí jeho zapsání do vlastnosti *Řádek nabídek* formuláře nebo sestavy.

9. Moduly (Lab06)

Programový modul je tvořen oblastí globálních definic a jednotlivých programových rutin (procedur a funkcí). K jejich tvorbě používáme jazyk *Visual Basic for Application*, který je společný všem produktům z rodiny Microsoft Office. Že se modul jako takový nespouští je zřejmé z již dříve uvedených informací o obsluze jednotlivých událostí vznikajících při práci s formuláři a sestavami.

Pro obsluhu událostí se dnes nejčastěji používají procedury událostí, shrnuté v modulu spojeném s obiektem, který obsluhují. Ve formuláři FProblemy nám chybí tlačítko pro přesun hotových problémů do zálohy. Potřebné tlačítko můžeme vytvořit pomocí průvodce s akcí *Různé* – Spustit dotaz. Potíž je v tom, že nevíme, zda to bude dotaz **OHotoveVvtvorit** nebo OHotovePresunout. Proto doplníme kód o zjištění, zda tabulka ProblemyHotove již existuje a podle toho se rozhodneme. Využíváme přitom objektu *Container*, který patří do struktury modelu DAO (Data Access Objects), který je uveden dále.



Obr. 60. Tvorba modulu

Private Sub OdeslatHotove_Click()

```
On Error GoTo Err_OdeslatHotove_Click
   Dim I As Long, Stat As Boolean
   Stat = False
   For I = 0 To
   DBEngine(0)(0).Containers("Tables").Doc
   uments.Count - 1
      Τf
   DBEngine(0)(0).Containers("Tables").Doc
   uments(I).Name = "ProblemyHotove" Then
          Stat = True
      End If
   Next T
   If Not Stat Then
      DoCmd.OpenQuery "QHotoveVytvorit",
   acNormal, acEdit
   Else
      DoCmd.OpenQuery "QHotovePresunout",
   acNormal, acEdit
   End If
   DoCmd.OpenOuery "OHotoveOdstranit",
   acNormal, acEdit
Exit OdeslatHotove Click:
   Exit Sub
Err OdeslatHotove Click:
   MsqBox Err.Description
   Resume Exit OdeslatHotove Click
```

End Sub

Strukturu objektů, seznam jejich vlastností a metod můžeme zjistit v okně prohlížeče objektů, viz obr. 61. Nabídka je rozdělena do skupin odpovídající jednotlivým knihovnám:

- Access objekty aplikace MS-Access, hlavní Application,
- DAO objekty databázového modelu, hlavní DBEngine,
- VBA objekty Visual Basic for Application,
- Lab01 objekty aktuální úlohy.

| 🚰 Prohlížeč objektů | | |
|---|-----------------------------|---------|
| <všechny knihovny=""></všechny> | 🖌 🔸 🖻 🎘 🤶 | |
| | ► ₩ × | |
| Třídy | Členové třídy "Application" | |
| 🗗 AcView 🔺 | as CurrentUser | |
| P AcWindowMode | 🔊 DAvg | |
| 🛃 Application | 🖻 DBEngine | |
| 💐 BoundObjectFrame | 🔊 DCount | |
| 🗗 CollatingOrderEnum | DDEExecute | |
| 🛃 Collection 📃 🚽 | as DDEInitiate | • |
| Property DBEngine As DBE jen pro čtení | ingine | 4 |
| Cien tridy <u>"Access</u> . <u>Applic</u> | ation | ⊡ |

Obr. 61. Prohlížeč objektů

V následujících podkapitolách jsou uvedeny jednoduché datové typy s postupem při deklarování proměnných a jednotlivé programové struktury pro řízení běhu programu. Rozsáhlejší popis vzhledem k rozsahu této příručky není možný. Stejně tak není uveden seznam funkcí ani seznamy metod jednotlivých tříd objektů.



Obr. 62. Objektový model MS-Access (MSACC8.OLB) a DAO (DAO350.DLL) pro ODBC Direct



Obr. 63. Objektový model DAO (DAO350.DLL) pro Microsoft Jet

Datové typy

V tabulce 9 jsou uvedeny názvy jednoduchých datových typů dostupných proměnným v programu. K jejich deklarování slouží příkaz Dim. Umožňuje kromě jednoduchých proměnných vytvářet také pole. Proměnné deklarované na úrovni modulu jsou dostupné všem procedurám v rámci modulu, proměnné deklarované na úrovni procedury jsou lokální.

Dim [**WithEvents**] *jméno*[([*rozměry pole*])] [**As** [**New**] typ]

Datový typ proměnné určujeme buď explicitně nebo použitím sufixu. Ten je sice součástí jména proměnné, ale nelze vytvořit dvě proměnné lišící se pouze sufixem.

Obdobně pracují příkazy **Private** pro deklaraci lokálních proměnných modulu a **Public** pro globální proměnné dostupné všem modulům, případně **Static** pro lokální proměnné procedury, která si zachová své hodnoty po celou dobu běhu programu.

| Datový typ | Velikost | Rozsah hodnot | Sufix |
|--------------|----------|--|-------|
| Boolean | 2 bajty | -1/0 (True/False) | |
| Byte | 1 bajt | 0 - 255 | |
| Integer | 2 bajty | -32 768 ÷ 32 767 | % |
| Long | 4 bajty | - 2 147 483 648 ÷ 2 147 483 647 | & |
| Single | 4 bajty | -3,402823E38 ÷ -1,401298E-45 pro záporná čísla | ! |
| | | 1,401298E-45 ÷ 3,402823E38 pro kladná čísla | |
| Double | 8 bajtů | -1,79769313486232E308 ÷ -4,94065645841247E-324 pro záporná čísla | # |
| | | 4,94065645841247E-324 ÷ 1,79769313486232E308 pro kladná čísla | |
| Currency | 8 bajtů | 922 337 203 685 477,5808 ÷ 922 337 203 685 477,5807 | @ |
| Date | 8 bajtů | datumy v rozsahu od 1. ledna 100 do 31. prosince 9999, časy od 0:00:00 do 23:59:59 | |
| String | | s proměnlivou délkou - až 2 miliardy (2^31) znaků | \$ |
| | | s pevnou délkou - od 1 znaku až do přibližně 64K (2^16) znaků | |
| Variant | | jakýkoliv datový typ nebo řetězec proměnné délky | |
| Decimal (jen | 14 bajtů | +/-79 228 162 514 264 337 593 543 950 335 bez desetinné čárky | |
| jako podtyp | | +/-7,9228162514264337593543950335 s 28 desetinnými místy | |
| Variant) | | nejmenší číslo různé od nuly je +/-0,00000000000000000000000000000000000 | |

Tab. 9. Jednoduché datové typy
Řízení běhu programu

Pro řízení běhu programu slouží následující struktury a příkazy:

| Do [{ While Until } <i>výraz</i>] | Do |
|---|----------------------------|
| tělo cyklu | tělo cyklu |
| Loop | Loop [{While Until} výraz] |

tělo cyklu se provádí pokud je splněna podmínka {While} nebo do splnění podmínky {Until} (za **Loop** je tomu opačně). Strukturu lze opustit příkazem **Exit Do** za příkaz **Loop**.

```
For počítadlo=start To cíl [Step krok]
tělo cyklu
Next [počítadlo[, vnořené počítadlo]]
```

tělo cyklu se provádí, dokud počítadlo nenabude hodnoty rovné nebo vyšší hodnotě cíl. Strukturu lze opustit příkazem **Exit For** za příkaz **Next**.

For Each *prvek* In *skupina* tělo cyklu Next [*prvek*]

tělo cyklu se provádí pro každý prvek pole nebo objekt zadané kolekce. Strukturu lze opustit příkazem **Exit For** za příkaz **Next**.

While *výraz* tělo cyklu Wend

tělo cyklu se provádí dokud má výraz hodnotu True.

| If výraz Then | If výraz 1 Then |
|---------------|---------------------|
| tělo cyklu 1 | tělo cyklu 1 |
| Else | ElseIf výraz 2 Then |
| tělo cyklu 2] | tělo cyklu 2 - 1 |
| End If | Else |
| | tělo cyklu 2 - 2] |

End If

Pokud má *výraz* hodnotu True, provede se tělo cyklu 1, jinak se provede tělo cyklu 2.

Select Case výraz Case hodnota 1 tělo 1 Case hodnota 2 tělo 2 [Case Else tělo když žádná hodnota] End Select

Podle hodnoty výrazu provede příslušnou činnost. Lze zadat seznam nebo rozsah hodnot (např.: 1 To 5; Is <15; Is >10).

GoSub řádek

Vyvolání procedury na určeném řádku. Řádek se označuje číslem nebo textovým návěštím odděleným dvojtečkou od vlastního řádku. Procedura končí příkazem **Return**.

On výraz GoTo když 1, když 2, když 3, ...

Podle hodnoty číselného výrazu pokračuje činnost na určených řádcích. Použití příkazu **Select Case** je ve většině případů výhodnější.

On výraz GoSub když 1, když 2, když 3, ...

Podle hodnoty číselného výrazu se vyvolá procedura na určených řádcích.

On Error {GoTo *řádek* | Resume Next | GoTo 0}

Určení, co se má provést v případě výskytu chyby. Příkaz **Resume Next** vrací řízení za příkaz, který vyvolal chybu, příkaz **Resume** na tento příkaz. Druh chyby a její systémové hlášení udávají vlastnosti objektu *Err - Err.Number* a *Err.Description*.

DoEvents()

Funkce DoEvents vrací číslo reprezentující počet otevřených formulářů. Funkce **DoEvents** předá řízení operačnímu systému. Řízení je vráceno jakmile operační systém dokončí obsluhu událostí své fronty a odešle všechny klávesy uvedené ve frontě **SendKeys**.

SendKeys řetězec[, wait]

Zasílá jedno nebo více stisknutí kláves do aktivního okna, jakoby byly stisknuty na klávesnici. Argument *wait* určuje, zda se řízení vrátí proceduře ihned po odeslání kláves (False) nebo až po jejich obsloužení (True).

Výpis struktury úkolů problému

Protože k vyřešení problému je třeba splnit několik úkolů a každý z nich může být podmíněn dílčími úkoly a ty opět dílčími úkoly, můžeme u složitých problémů ztrácet přehled o návaznosti úkolů. Struktura úkolů tvoří strom s předem neomezeným počtem úrovní, proto není možné vytvořit za tímto účelem sestavu. K zobrazení stromu úkolů vytvoříme funkci *Strom*, jejíž algoritmus je uveden na obr. 64. Ta postupně zpracuje všechny hlavní úkoly. Ke zpracování jednoho problému použije funkci *StromZpracuj*, která pro zpracování podúkolů používá opět funkci *StromZpracuj* (rekurzívní volání).

Pro zobrazení stromu použijeme textový dokument, který za tímto účelem vytvoříme. Umístíme ho do stejné složky, jako aktuální databázi. Za účelem určení aktuálního umístění byla přidána funkce *Pracovni_adresar*. Pro případ víceuživatelského přístupu k databázi a tedy možnosti současného spuštění je třeba nejprve najít nepoužívaný název dokumentu. Po vytvoření dokument otevřeme a po přečtení ihned zrušíme, aby se nám na disku nehromadily staré dokumenty.

Pro spuštění funkce *Strom* přidáme do formuláře *FProblemy* další příkazové tlačítko.



Obr. 64. Algoritmus zpracování stromu úkolů

```
Option Compare Database
Option Explicit
```

' Funkce pro výpis struktury problémů a úkolů
' Copyright © 1999, Ing. Radim Farana, CSc.

Private Function Pracovni_adresar()

```
. ******
' funkce vrátí název cesty k aktuální databázi
Dim D As Database, S As String
Dim I As Integer, J As Integer
Set D = DBEngine.Workspaces(0).Databases(0)
S = D.Name
Pracovni adresar = "."
If Len(S) > 0 Then
   -T = Ο
   For I = Len(S) To 1 Step -1
      If J = 0 And Mid(S, I, 1) = "\" Then
         J = I
      End If
   Next
   Tf J > 0 Then
      S = Left(S, J - 1)
      Pracovni adresar = S
   End If
End If
End Function
```

```
Public Function Strom(Problem As Variant)
. ****************
' funkce vygeneruje strukturu úkolů k vyřešení
   problému
. *******
' deklarace proměnných
Dim Cis As Variant, Poc As Integer, Naz As String,
   S As String
Dim D As Database, R As Recordset, N As Long
' kontrola zadání vstupního parametru
If IsNull(Problem) Or Problem = "" Then
   MsqBox "Nebyl zadán požadovaný problém, není
   co zobrazit.", vbCritical, "Chyba výpisu
   problému"
Else
    ' inicializace výstupního dokumentu
   Cis = FreeFile
   Poc = 1
   Naz = ""
   Do While Naz = ""
       S = Dir(Pracovni adresar() & "\Prob" &
   Trim(Str(Poc)) & ".TXT")
       Tf S = "" Then
          Naz = Pracovni adresar() & "\Prob" &
   Trim(Str(Poc)) & ".TXT"
       Else
           Poc = Poc + 1
       End If
   Loop
   Open Naz For Output Access Write As #Cis
   Print #Cis, "Výpis úkolů problému: " &
   DFirst("Nazev", "Problemy", Problem)
```

```
Print #Cis,
    ' zjištění množiny hlavních úkolů
   Set D = DBEngine(0)(0)
   Set R = D.OpenRecordset("SELECT * FROM Ukoly
   WHERE (([IDProblem]=" & Problem & ") And
   ([IDMaster]=[ID])) ORDER BY DatumOd,
   DatumDo;", dbOpenSnapshot)
   ' nastavení úrovně vnoření do stromu
   \mathbf{N} = \mathbf{0}
   ' zpracování kořenových prvků
   While Not R.EOF
       Print #Cis, Space(2 * N); "- "; R![Osoba]
   & " : " & R![DatumOd] & " - " & R![DatumDo]
       Call StromZpracuj(R![ID], N, Cis)
       R.MoveNext
   Wend
   ' ukončení výstupního dokumentu
   Close #Cis
   S = Str(Shell("Notepad.exe " & Naz, 1))
   MsgBox "Byl vygenerován dokument '" & Naz & "'
   s výpisem úkolů. Po potvrzení tohoto hlášení
   bude dokument zrušen.", vbInformation, "Konec
   výpisu"
   S = Dir(Naz)
   If Len(S) > 0 Then
       Kill Naz
   End If
End If
End Function
```

```
Private Sub StromZpracuj(ID As Long, N As Long,
   Cis As Variant)
. ******
' funkce zpracuje jeden úkol včetně podúkolů
Dim D As Database, R As Recordset
' výběr potomků pro zpracování
Set D = DBEngine(0)(0)
Set R = D.OpenRecordset("SELECT * FROM Ukoly WHERE
   (([IDMaster]=" & ID & ") And
   Not([IDMaster]=[ID])) ORDER BY DatumOd,
   DatumDo;", dbOpenSnapshot)
' zvýšení hloubky vnoření
N = N + 1
' zpracování potomků
While Not R.EOF
   Print #Cis, Space(2 * N); "- "; R![Osoba] & "
   : " & R![DatumOd] & " - " & R![DatumDo]
   Call StromZpracuj(R![ID], N, Cis)
   R.MoveNext
Wend
' návrat na původní hloubku vnoření
```

```
N = N - 1
End Sub
```

10. Bezpečnost systému (Lab07)

V řadě úloh vyžadujeme rozlišení přístupových práv jednotlivých oprávněných uživatelů a současně zamezení přístupu neautorizovaných uživatelů k datům.

Základem úspěchu je zamezení možnosti číst databázový dokument v otevřené podobě. Při troše dovedností by nebyl problém získat všechna uložena data, jak vidíme z úryvku dokumentu:

...@Kubikula, KubaTvorba a propojení
formulářůG+_____`łá@____`ťá@Kubikula,
KubaNávrh a realizace
dotazůC+______ŕłá@____´á@Ko
trba, MatějTvorba vazeb mezi tabulkami,
referenční integrita ...

K tomu použijeme zašifrování databáze, které systém obsahuje. Používá se algoritmus RC4 of firmy RSA Data Security Inc., což je proudová šifra používající 32 bitový klíč na 2 kB datovou stránku. Šifrovací klíč (neboť se jedná o systém s tajným klíčem) se generuje z čísla pracovní skupiny, jak uvidíme níže.

Dále potřebujeme vytvořit účty jednotlivých uživatelů a určit rozsah jejich přístupových práv k systému. Potřebné údaje o uživatelích se ukládají do informačního souboru pracovní skupiny. Vytvoříme ho pomocí aplikace *Administrátor pracovních skupin*. Obvykle je k dispozici ve složce Windows\System:

C:\WINDOWS\SYSTEM\WRKGADM.EXE

Má jen několik nabídek, viz obr. 65. Pro nás je nejdůležitější vytvoření nové pracovní skupiny PROBLEM.MDW. Postupně zadáme:

- údaje o vlastníkovi pracovní skupiny a hlavně identifikační číslo skupiny (používá se jako heslo pro šifrování databáze, s jeho znalostí je možno vytvořit opakovaně totožný soubor pracovní skupiny, proto je třeba ho velmi dobře utajit),
- 2. cestu a název nového souboru pracovní skupiny,
- 3. potvrdit zadaná data.

K nově vytvořené pracovní skupině budeme ihned také připojeni.



Obr. 65. Administrátor pracovních skupin

| Informace o vlastníkovi pracovní skupiny 🔀 | | | | | |
|--|--------------------------|--|--|--|--|
| Nový informační soubor pracovní skupiny je určen jménem, názvem firmy a identifikátorem pracovní skupiny, ve kterém se rozlišují veľká a malá písmena. | | | | | |
| Použijte níže uvedené jméno a název firmy, příp. zadejte odlišné jméno či firmu. Chcete-li zajistit, aby vaše pracovní skupina byla jedinečná, zadejte jedinečnou identifikaci skupiny obsahující maximálně 20 čísel nebo znaků. | | | | | |
| <u>J</u> méno: | Radim Farana | | | | |
| <u>F</u> irma: | VŠB-TU Ostrava | | | | |
| ID pracovní skupiny: | piny: PBgroup789atre&7Dc | | | | |
| | OK Storno | | | | |

Obr. 66. Vytvoření nové pracovní skupiny

Nová pracovní skupina obsahuje několik uživatelů a skupin. Po přihlášení k úloze doplníme pracovní skupiny:

- *uživatelé* systémová skupina, všichni uživatelé musí být jejími členy,
- administrátoři systémová skupina,
- *čtenáři* PID: GR91opc,
- *hosté* PID: GR92utb.

Obdobně vytvoříme potřebné uživatele, viz obr. 67:

- *administrátor* implicitní vlastník systému, také aktuálně přihlášený uživatel,
- *host* PID: U112cs, (skupina hosté),
- *čtenář* PID: U196ci, (skupina čtenáři),
- správce PID: U220sk, (skupina administrátoři) – nový vlastník systému.

| Účty uživatelů a skupin 🔹 🔀 |
|--|
| Uživatelé Skupiny Změnit heslo |
| Uživatel Názgv: host Nový Odstranit Vymazat heslo Členství ve skupinách Dostupné skupiny: Člen skupiny: administrátoři Čtenáři hosté uživatelé << Odstranit |
| ⊻ytisknout uživatele a skupiny |
| OK Storno Použít |

Obr. 67. Správa účtů uživatelů a skupin

Nyní jsme již připraveni použít *Průvodce uživatelskou úrovní zabezpečení*, viz. obr. 68. Je třeba velmi přesně dodržet následující postup:

1. Aktuálnímu uživateli (*administrátor*, anglicky admin) přidělíme heslo (admin), toto heslo není nutno chránit před prozrazením, pokud následně odebereme tomuto uživateli všechna práva. Pokud nemá administrátor přiřazeno heslo, hlásí se k úloze automaticky všichni uživatelé jako *administrátor*.

- 2. Uzavřeme databázi a znovu se do ní přihlásíme jako *správce*. Bude vhodné přidělit mu heslo.
- Spustíme *Průvodce uživatelskou úrovní zabezpečení* dle obr. 68 a databázi převedeme na zabezpečenou databázi. Původní nezabezpečená databáze zůstane nezměněna.
- 4. Nově vytvořenou databázi otevřeme jako uživatel *správce*. Ten je nyní vlastníkem všech objektů a jako jediný má všechna práva k objektům.
- 5. Nyní by podle nápovědy měla být zrušena skupina administrátoři a uživatel administrátor. Záhy zjistíme, že to není možné, proto pouze odebereme administrátoru všechna práva ke všem objektům a zrušíme jeho členství ve skupině administrátoři.
- Upravíme práva jednotlivých skupin (obr. 69): *administrátoři* – všechna práva *uživatelé* – žádná práva *čtenáři* – čtení návrhu a všechny manipulace s daty, *hosté* – čtení návrhu a čtení dat.
- 7. Posledním krokem je vytvoření zkomprimovaného databázového dokumentu MDE, který bude zpřístupněn uživatelům.

Pozn.: Po vytvoření databázového dokumentu MDE je třeba ho otevřít, zobrazit skryté objekty a skupině hosté přidělit práva k tabulce **MSysModules** pro **Aktualizaci dat**. Jinak bude jejím členům při vstupu do systému hlášena chyba – nedostatek práv k této tabulce.

| 🗃 Průvodce uživatelskou úrovní zabezpečení 🛛 🔀 | | | | |
|---|-----------------|---------------------|----------------|--|
| Průvodce vytvoří novou zab databázi nebudou proveden problematiky zobrazíte klepn | (OK) Storno | | | |
| _ Které objekty mají být zabe | Nápověda | | | |
| 🔽 🔤 Iabulky 🔽 🗹 | Eormuláře | 🗹 <u>M</u> akra | | |
| ☑ Dotazy | <u>S</u> estavy | Moduly | | |
| Aktuální uživatel: | Správce | | | |
| Soubor pracovní skupiny: | D:\Users\FAR10 | \Dbacc97\Cviceni\La | b07\Problem.md | |

Obr. 68. Průvodce uživatelskou úrovní zabezpečení

| Práva uživatelů a skupin | ? × | | |
|------------------------------------|--|--|--|
| Uživatelská práva Změnit vlastníka | | | |
| Náz <u>e</u> v uživatele/skupiny: | Název objektu: | | |
| administrátor Čtenář | <nové dotazy="" tabulky=""> Osoby</nové> | | |
| host Správce | Problemy ProblemyHotove | | |
| | Ukoly | | |
| | | | |
| Seznam: 💽 Uživatelé 🔘 Skupiny | Iyp objektu: tabulka 💌 | | |
| – Uživatelská práva – | | | |
| Ctevírání/spouštění | 🔽 Čtení <u>d</u> at | | |
| ✓ Čtení <u>n</u> ávrhu | ✓ Aktualizace dat | | |
| 🔽 Změna návrhu | 🔽 Vkládání dat | | |
| 🔽 Sprá <u>v</u> a | ✓ Odstraňování dat | | |
| | | | |
| Aktualni uzivatei: | | | |
| OK | Storno P <u>o</u> užít | | |

Obr. 69. Přidělení práv k objektům

Pro spouštění úlohy je nyní vhodné vytvořit zástupce (obr. 70), který umožní zapsat spuštění úlohy se všemi potřebnými parametry:

```
D:\MSOffice\Office\MSACCESS.EXE
```

```
D:\Users\FAR10\Dbacc97\Cviceni\Lab07\La
b07Sol.mde /wrkgrp
D:\Users\FAR10\Dbacc97\Cviceni\Lab07\Pr
oblem.mdw /nostartup
```

Skládá se z:

- cesty k programu MS-Access,
- cesty k úloze (soubor MDE),
- parametru *wrkgrp* a cesty k souboru pracovní skupiny,
- parametru *nostartup*, který zakazuje zobrazení úvodního okna systému.

Pokud máme k dispozici *Microsoft Office 97 Developers Edition Tools*, můžeme vytvořit také instalační diskety (nebo síťovou instalaci systému), včetně doplnění o *Run-Time* modul, takže uživatel nemusí být vlastníkem MS-Access.

Spustíme aplikaci *Setup Wizard* a budeme provedeni všemi potřebnými nastaveními:

- 1. Vybereme všechny soubory, které chceme do instalace zařadit a určíme jejich základní zařazení, viz obr. 71.
- 2. Vytvoříme všechny požadované zástupce a popíšeme jejich určení a obsah (parametry pro spuštění úlohy),



Obr. 70. Zástupce pro spouštění úlohy

- 1. Definujeme položky, které mají být zapsány do databáze Registry systému Windows po instalaci naší úlohy.
- 2. Zvolíme instalované komponenty (Run-Time, Workgroup Administrator, Replication Manager, podpora ODBC a další).
- 3. Vybereme požadované způsoby instalace (Typical, Compact, Custom).
- 4. Určíme název aplikace, číslo verze a cílovou složku pro instalaci.
- 5. Určíme, zda se po dokončení instalace má otevřít nějaký dokument (ReadMe apod.).
- 6. Určíme způsob distribuce instalace systému (diskety, síťová instalace) a instalaci spustíme.

Pozn.: Pokud očekáváme opakování tvorby instalace systému, není špatné si těsně před vlastním provedením definici uložit. Systém vytvoří speciální dokument MDT. Ve skutečnosti se jedná o databázový dokument MS-Access s několika tabulkami, obsahujícími data pro instalátor.

| Setup Wizard Add the files that you wa | nt your custom Setup program to copy and then se | t properties for each file: |
|---|---|--|
| List of Files: D:\Users\FAR10\Dbacc9 D:\Users\FAR10\Dbacc9 D:\Users\FAR10\Dbacc9 | /\Cviceni\Lab07\Lab075ol.mde /\Cviceni\Lab07\PROBLEM.MDW /\Cviceni\Lab07\TUTORIAL.ICO | Add Delete |
| File Properties: | | |
| File Name and Path: | D:\Users\FAR10\Dbacc97\Cviceni\Lab07\Lab07Sc | ol.mde B <u>r</u> owse |
| Destination Folder: | \$(AppPath) | Set as Application's Main File |
| Overwrite Existing File: | Older | Include File on First Disk ✓ Compress File |
| Component Name: | Application | Set as Workgroup File |
| Help Sav | re Cancel < Bar | ck <u>N</u> ext > ⊟nish |

Obr. 71. Tvorba instalačních disket (Setup Wizard)

11. Replikace (Lab08)

Replikace databází je technologie alternativní k *distribuovaným databázovým systémům*. Místo rozdělení dat na několik míst a zapojení systému, který požadovaná data prezentuje uživateli, se všechna data umístí na všechna místa. Vzniká sice značná duplicita, ale zato mají všichni uživatelé okamžitý přístup ke všem datům. Jednotlivé instance přitom žijí svým nezávislým životem a data v nich se brzy začnou lišit. Je nutné provedení *synchronizace*.



Obr. 72. Princip replikace databází

Základem systému replikace je databáze označená jako *řídicí vzor*. Ta má právo předat ostatním účastníkům nejen data, ale také změny návrhu jednotlivých objektů. Ostatní účastníci jsou

z *řídicího vzoru* vygenerováni a nazývají se *repliky*. Ty mohou předávat pouze změny dat. Strategie provedení synchronizace závisí na dislokaci replik a možnosti jejich vzájemného spojení. Synchronizace totiž probíhá vždy mezi dvěma účastníky. Pro hvězdicovou strukturu dle obr. 72 musí být provedeno pět synchronizací, aby se do všech replik promítly všechny změny dat.

Vytvoření řídicího vzoru a repliky z připraveného databázového dokumentu je velmi jednoduché. Z nabídky *Nástroje-Replikace* zvolíme *Vytvořit repliku*. Systém provede následující úkony:

- nabídne vytvoření zálohy aktuálního dokumentu,
- převede aktuální databázový dokument na řídicí vzor (doplní potřebné systémové tabulky a replikační identifikátory),
- vytvoří jednu repliku požadovaného jména.

Nyní je třeba pravidelně provádět synchronizaci, viz obr. 73. V našem případě ji budeme spouštět z řídicího vzoru.



Obr. 73. Spuštění synchronizace

Výsledkem synchronizace může být hlášení o úspěšném ukončení synchronizace nebo o vzniku konfliktů (velmi snadno nastane pokud v obou dokumentech byl změněn týž záznam a synchronizace je vyvolána z prostředí repliky). V tom případě můžeme ihned přistoupit k vyřešení konfliktů. Systém vypíše seznam tabulek s konflikty, viz obr. 74. Systém pro každou tabulku vygeneruje automatický formulář a nabídne možnost ponechat existující záznam nebo jej nahradit konfliktním záznamem ze synchronizovaného dokumentu.

| Vyřešit konflikty v replikách | | | |
|-------------------------------|---|--|--|
| s konflikty replik: | | | |
| y 1 Chyby <u>n</u> ávrhu | 1 | | |
| Chyby <u>d</u> at | 1 | | |
| Zavřít | 1 | | |
| Chyby <u>dat</u> Zavřít | | | |

Obr. 74. Vzniklé konflikty při synchronizaci

| 📰 Řešení konflik | tů při replikaci - Osoby | | × |
|------------------|----------------------------|--------------------------------------|---|
| | Existující záznam | Konfliktní záznam | |
| Osoba: | Babík, Jan | Babík, Jan, Václav | |
| | Zachovat existující záznam | <u>P</u> řepsat konfliktním záznamem | |
| Záznam: 🚺 🕢 | 1 🕨 🕨 🕨 z 1 | | |

Obr. 75. Automatický formulář pro řešení vzniklých konfliktů

Pro podporu replikace systému je možno využít aplikaci *Replication Manager*, která je součástí *Microsoft Office 97 Developers Edition Tools*, viz obr. 76.



Obr. 76. Replication Manager

Aplikace umožňuje vytváření replik i jejich údržbu. Pokud je spuštěn, umožňuje také automatické spouštění synchronizace v určených časech, viz obr. 79. Systém také udržuje informační dokument (LOG) s informacemi o své činnosti, viz obr. 80.

Řešení vzniklých konfliktů však zůstává na uživatelích jednotlivých replik.

| Properties | <u>? ×</u> |
|--|--|
| Replica Set | Synchronizer |
| Open Database | Managed Replicas |
| Members of this set at 'Synchro | onizer': |
| D:\Users\FAR10\Dbacc97\C D:\Users\FAR10\Dbacc97\C | viceni\Lab08\Lab08.mdb .viceni\Lab08\Lab08Rep.mdb |
| ✓ Is the Design Master ✓ Is a Partial Replica | 🗖 is Read Only |

Obr. 77. Vlastnosti replikace

| Modify Object Status | ? × |
|--|---|
| Tables Queries Forms Reports Macros Replicated objects are available to the entire replica set, while local Making an object replicated or local can be done only at the Design Beplicated Tables: Page 100 and 100 an | Modules objects are not. Master. |
| ✓ Osoby ✓ Problemy ✓ ProblemyHotove ✓ Ukoly | Select <u>A</u> ll Select <u>N</u> one |
| OK Cancel Apply Now | |

85

Edit Locally Managed Replica Schedule

The Synchronizer uses this schedule to synchronize all locally managed members of a replica set. The same schedule is used for all replica sets managed by this Synchronizer.



Obr. 79. Plán automatické synchronizace

| ync | hronization Histor | y | | | ? × |
|--------------|---|-------------|----------|--------------------------------|---------------|
| | Synchroniza | ition betwe | en D:\U: | sers\FAR10\Dbacc97\Cviceni\Lal | b08\Lab08.mdb |
| and | and replica set members managed by This Synchronizer (Synchronizer) | | | | |
| | Filter | history usi | ng: Last | 25 Synchronizations | • |
| | Date Initiated | Records | Problems | Reason for Failure | Close |
| - | 12.9.1999 12:38:52 | 8 | 0 | | |
| - | 12.9.1999 12:38:29 | 1 | 0 | | Details |
| \checkmark | 12.9.1999 12:36:36 | 0 | 0 | | <u></u> |
| - | 12.9.1999 12:28:35 | 0 | 0 | | |
| - | 11.9.1999 15:00:07 | 0 | 0 | | |
| - | 11.9.1999 14:54:40 | 1 | 1 | | |
| < | 11.9.1999 14:53:12 | 9 | 1 | | |
| - | 11.9.1999 14:52:27 | 1 | 0 | | |
| - | 11.9.1999 14:52:09 | 8 | 0 | | |
| ~ | 11.9.1999 14:51:26 | 8 | 1 | | |
| < | 11.9.1999 14:46:30 | 0 | 0 | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| • | | | | • | |
| | | | | | |

Obr. 80. Informace o výsledcích synchronizací

12. Spolupráce aplikací (Lab09)

Již bylo zmíněno, že všechny aplikace z rodiny MS-Office používají jednotný jazyk *Visual Basic for Applications*. To nám umožňuje jednoduše realizovat spolupráci jednotlivých aplikací. V našem případě se budeme snažit o přístup k datům z prostředí MS-Excel.

K dispozici jsou dvě možnosti přístupu k úloze. Buď využijeme přímo objekt knihovny DAO DBEngine, tedy databázové jádro Jet Engine, nebo konektujeme databázi pomocí ODBC Direct (*Open DataBase Connectivity*). My vyzkoušíme oba způsoby. Pro zadání potřebných parametrů vytvoříme v prostředí MS-Excel dokument, viz obr. 81. Všimněte si zejména pojmenování jednotlivých polí, na která se budeme v programu odkazovat.

Tvorba potřebných funkcí se provádí v okně *Nástroje-Makro-Editor jazyka Visual Basic*, viz obr. 82.

| ХМ | 🗙 Microsoft Excel - Lab09Sol.xls | | | | |
|--|---|---|--------------------|--|--|
| 8 | Soubor Úpr <u>a</u> vy Zobrazit V <u>l</u> ožit | Eormát <u>N</u> ástroje <u>D</u> ata <u>O</u> kno Nápo <u>v</u> ěda | _ 8 × | | |
| | 🛎 🖬 🍯 🖪 🖤 👗 🖻 | 💼 ダ 🗠 - ལ - 🍓 💝 🗵 🏂 🛓 👪 🖉 🚜 | 100% 🔹 | | |
| Aria | al CE 🔹 10 💌 🖪 | 3 / U 三三三國 🕄 % , 👷 🦉 🖽 | • 🕭 • <u>A</u> • 炎 | | |
| | SystemDB 🗾 = D: | \Users\FAR10\Dbacc97\Cviceni\Lab07\PROBLEM.MDW | | | |
| | A | В | C 🛓 | | |
| 1 | Ukázka spojení ap | likace z MS-Excel do MS-Access | | | |
| 2 | enazia ekeleni ak | | | | |
| 3 | Datový soubor: 🛛 ==> | D:\Users\FAR10\Dbacc97\Cviceni\Lab07\Lab07.mdb | FileName | | |
| 4 | | GO | | | |
| 5 | Sub UseDBEngine() | | | | |
| 6 | Pomocí DBEngine Ize kontak | tovat pouze nezabezpečenou databázi. | | | |
| 7 | Není možno určit systémovou | u databázi s právy uživatele, vždy se používá standardní: | | | |
| 8 | C:\WINDOWS\SYSTEM\SYS | TEM.MDW | | | |
| 9 | | | | | |
| 10 | Parametry ODBC: | | | | |
| 11 | Datový soubor:> | D:\Users\FAR10\Dbacc97\Cviceni\Lab07\Lab07Sol.mde | DataPath | | |
| 12 | Název aplikace: | Lab07Sol | ODBCName | | |
| 13 | Jméno uživatele: | Čtenář | ODBCUser | | |
| 14 | Jeho heslo: | | ODBCPWD | | |
| 15 | Systémová databáze: | D:\Users\FAR10\Dbacc97\Cviceni\Lab07\PROBLEM.MDW | SystemDB | | |
| 16 | | | | | |
| 17 | Kontakt databáze pomocí OD | BC GO | | | |
| 18 | (Pokud není definován zdroj, p | rovede jeho definici) | | | |
| 19 | Sub UseODBC() | | | | |
| R I | Database | | | | |
| 📗 <u>K</u> reslení 🔹 😓 🌝 🛛 Automatické tvary 🔹 🛝 🗔 📿 🚰 🚚 🤌 🗉 🚄 🖌 📥 🖛 🛱 🛄 | | | | | |
| Připr | raven | 123 | | | |

Obr. 81. Dokument MS-Excel

| 🚈 Microsoft Visual Basic - Lab09Sol.xls [přerušit] | | | | |
|--|--------------------|--|--|--|
| Soubor Úpr <u>a</u> vy Zobrazit Vložit | <u>F</u> ormát Lag | íní Sp <u>u</u> stit <u>N</u> ástroje <u>O</u> kno Nápo <u>v</u> ěda | | |
| 🛛 🗙 🛅 • 🖬 🖌 🗛 🖪 🖊 | юœ | 🕨 📋 🖬 👹 😭 📽 🎌 🔯 ř. 9, sl. 9 | | |
| Projekt - VBAProject 🔀 | 🐹 Lab09 | ol.xls - ThisWorkbook (kód) | | |
| | (obecné | ▼ UseDBEngine | | |
| WBAProject (Lab0950) Microsoft Excel Objek Microsoft Excel Objek Wiastnosti - ThisWorkbook Vlastnosti - ThisWorkbook Vlastnosti - ThisWorkbook Abecedně Podle kategorií (Name) ThisWorkbook AcceptLabelsInf True AutoUpdateFrec ConflictResolutic 1 Date1904 False DisplayDrawingC HighlightChange False ChangeHistoryD Lich | Pub | <pre>Lic Sub UseDBEngine() 'Pro spuštění je nutno zajistit přístup k DAO (De 'V nabídce Nástroje-Odkazy-Microsoft DAO 3.5 Obje Dim W As Workspace, D As Database, R As Recordse Dim DM As Date, S As String, T As String On Error GoTo Err_UseDBEngine 'Otevření pracovního prostoru a připojení databá S = ThisVorkbook.Worksheets("DataBase").Range("F If S = "" Or IsNull(S) Then Exit Sub Set W = DAO.DBENGINE.CreateWorkspace("JetWorkspace Set D = W.OpenDatabase(S) 'Určení aktuálního data DM = Date 'Zjištění seznamu úkolů, které mají být dnes řeše Set R = D.OpenRecordset("SELECT QProblemyUkoly.* T = ""</pre> | ata Access Object) ect library t ze (Pozor Excel pr ileName") (1, 1).Te ce", "Admin", "", eny FROM QProblemyUko | |
| KeepChangeHist True | | | | |
| ListChangesOnly False | Kukátka | | × | |
| PersonalViewList True | Výraz | Hodnota Typ | Kontext 🔺 | |
| PersonalViewPrir True | 66 DM | 0:00:00 Date 1 | hisWorkbook.UseDBEngine | |
| PrecisionAsDispl False | 66 S | "D:"Users1FAR1U/Dbacc971CVI String | nisyvorkbook.UseDBEngine | |
| Saved True | 00 | String | riisvvorkbook.osebbengine | |
| SaveLinkValues frue | | | | |
| | | | | |
| | | | _ | |

Obr. 82. Tvorba programových rutin Visual Basic for Application

```
Public Sub UseDBEngine()
    'Pro spuštění je nutno zajistit přístup k DAO
    (Data Access Object)
    'V nabídce Nástroje-Odkazy-Microsoft DAO 3.5
    Object library
    Dim W As Workspace, D As Database, R As
    Recordset
    Dim DM As Date, S As String, T As String
    On Error GoTo Err UseDBEngine
    'Otevření pracovního prostoru a připojení
    databáze (Pozor Excel pracuje ve složce
    Office!)
    S =
    ThisWorkbook.Worksheets("DataBase").Range("Fil
    eName")(1, 1).Text
    If S = "" Or IsNull(S) Then Exit Sub
    Set W =
    DAO.DBENGINE.CreateWorkspace("JetWorkspace",
    "Admin", "", dbUseJet)
    Set D = W.OpenDatabase(S)
    'Určení aktuálního data
    DM = Date
    'Zjištění seznamu úkolů, které mají být dnes
    řešenv
    Set R = D.OpenRecordset("SELECT
    OProblemyUkoly.* FROM OProblemyUkoly WHERE
    [DatumOd] <= " & MakeSOLDate(DM) & " AND
    [DatumDo]>=" & MakeSQLDate(DM) & " ORDER BY
    DatumOd, DatumDo;", dbOpenForwardOnly)
    T = ""
    Do Until R.EOF
        If Not T = "" Then
            T = T \& Chr(13) \& Chr(10)
        End If
```

T = T & R![Osoba] & " : " &R![DatumOd] & " - " & R![DatumDo] & " : " & R![Reseni] R.MoveNext Toop 'Uzavření otevřených objektů D.Close W.Close 'Výpis zjištěných údajů MsqBox T, vbInformation, "Dnešní úkoly: " & DM Exit Sub Err UseDBEngine: MsgBox Error, vbCritical, "Error" End Sub Public Sub UseODBC() Dim W As Workspace, D As Connection, R As Recordset, R1 As Recordset Dim DM As Date, TM As Date, S As String, T As String, I As Long Dim S1 As String, S2 As String, S3 As String, S4 As String, SP As String Dim Pokus As Boolean On Error GoTo Err UseODBC 'Otevření pracovního prostoru a připojení databáze pomocí ODBC S = ThisWorkbook.Worksheets("DataBase").Range("Dat aPath")(1, 1).Text S1 = ThisWorkbook.Worksheets("DataBase").Range("ODB CName")(1, 1).Text If S1 = "" Or IsNull(S1) Then S1 = "Lab06Sol"

```
S2 =
ThisWorkbook.Worksheets("DataBase").Range("ODB
CUser")(1, 1).Text
If S2 = "" Or IsNull(S2) Then S2 = "Admin"
S3 =
ThisWorkbook.Worksheets("DataBase").Range("ODB
CPWD")(1, 1).Text
If S3 = "" Or IsNull(S3) Then S3 = ""
S4 =
ThisWorkbook.Worksheets("DataBase").Range("Sys
temDB")(1, 1).Text
If S4 = "" Or IsNull(S4) Then S4 =
"PROBLEM.MDW"
Set W =
DAO.DBENGINE.CreateWorkspace("JetWorkspace",
"admin", "", dbUseODBC)
S = "ODBC;DSN=" & S1 & ";" & "UID=" & S2 &
";PWD=" & S3 & ";DATABASE=" & S
Set D = W.OpenConnection(S1, dbDriverNoPrompt,
False, S)
'Určení aktuálního data
DM = Date
'Zjištění seznamu úkolů, které mají být dnes
řešeny
Set R = D.OpenRecordset("SELECT
OProblemyUkoly.* FROM OProblemyUkoly WHERE
[DatumOd] <= " & MakeSOLDate(DM) & " AND</pre>
[DatumDo]>=" & MakeSOLDate(DM) & " ORDER BY
DatumOd, DatumDo;", dbOpenForwardOnly)
Т = ""
Do Until R.EOF
    If Not T = "" Then
        T = T \& Chr(13) \& Chr(10)
```

```
End If
           T = T \& R![Osoba] \& " : " \&
   R![DatumOd] & " - " & R![DatumDo] & " : " &
   R![Reseni]
       R.MoveNext
   Loop
    'Výpis zjištěných údajů
   MsqBox T, vbInformation, "Dnešní úkoly: " & DM
    'Uzavření otevřených objektů
End UseODBC:
   D.Close
   W.Close
Exit UseODBC:
   Exit Sub
Err UseODBC:
   If Pokus = True Then
       MsqBox Err.Description, vbCritical,
    "Nedaří se zaregistrovat ODBC zdroj"
       Resume Exit UseODBC
   Else
        Select Case Err
            Case Is = 3146
             'Neúspěšné volání ODBC, pokusíme se
    zaregistrovat zdroj
             SP =
   ThisWorkbook.Worksheets("DataBase").Range("Dat
   aPath")(1, 1).Text
             If SP = "" Or IsNull(SP) Then Resume
    Exit UseODBC
            If Dir(SP) = "" Then
                'Nedaří se najít datový soubor,
   registrace nemá smysl
```

```
MsqBox "Není k dispozici datový
zdroj: " & SP & vbCr & "registrace ODBC zdroje
není možná!", vbCritical, "Definice ODBC"
           Resume Exit UseODBC
        End If
        T = 0
        While InStr(I + 1, SP,
Application.PathSeparator) > 0
           I = InStr(I + 1, SP)
Application.PathSeparator)
        Wend
        T = Left(SP, I - 1)
        SP = "Description=Cvičná databáze" &
vbCr &
           "DefaultDir=" & T & vbCr &
           "Driver=Microsoft Access Driver
(*.mdb)" & vbCr & _
           "DBO=" & SP & vbCr &
           "FIL=MS Access" & vbCr &
           "SystemDB=" & S4 & vbCr &
           "MaxBufferSize=512" & vbCr & _
           "PageTimeout=600" & vbCr &
           "UTD=" & S2 & ";"
        'Registrace DSN pro ODBC ovladač MS
Access
       DBENGINE.RegisterDatabase S1,
"Microsoft Access Driver (*.mdb)", True, SP
       Pokus = True
        'Návrat k připojení datového zdroje
       Resume
    Case Else
       MsqBox Err.Description, vbCritical,
"Error"
       Pokus = False
```

Resume Exit UseODBC End Select End If End Sub Public Sub DataPathSet() Dim S As Variant, R As Range ThisWorkbook.Worksheets("DataBase").Unprotect S = Application.GetOpenFilenameIf S = False Or IsNull(S) Or S = "" ThenMsqBox "Cesta k datovému souboru nebyla zadána!", vbCritical, "STOP" Else 'Kontrola přípony If LCase(Right(S, 4)) = ".mdb" Or LCase(Right(S, 4)) = ".mde" Then Set R = ThisWorkbook.Worksheets("DataBase").Range("Dat aPath") R(1, 1).Value = S Else MsqBox "Je třeba vybrat databázový soubor MS-Access (*.mdb, *.mde)!", vbExclamation, "STOP" End If End If ThisWorkbook.Worksheets("DataBase").Protect End Sub Public Sub FileNameSet() Dim S As Variant, R As Range ThisWorkbook.Worksheets("DataBase").Unprotect S = Application.GetOpenFilename If S = False Or IsNull(S) Or S = "" Then MsqBox "Cesta k datovému souboru nebyla zadána!", vbCritical, "STOP"

```
Else
        'Kontrola přípony
        If LCase(Right(S, 4)) = ".mdb" Or
    LCase(Right(S, 4)) = ".mde" Then
            Set R =
    ThisWorkbook.Worksheets("DataBase").Range("Fil
    eName")
            R(1, 1).Value = S
        Else
            MsqBox "Je třeba vybrat databázový
    soubor MS-Access (*.mdb, *.mde)!",
    vbExclamation, "STOP"
        End If
    End If
    ThisWorkbook.Worksheets("DataBase").Protect
End Sub
Public Function MakeSQLDate(D As Date) As String
If IsNull(D) Then
    MakeSOLDate = "Null"
Else
    MakeSOLDate = "#" & Month(D) & "/" & Day(D) &
    "/" & Right(Str(Year(D)), 2) & "#"
End If
End Function
```

U funkce *UseODBC()* využíváme pro přístup k databázi ODBC. Pokud se nepodaří požadované spojení najít, v rámci obsluhy vzniklé chyby se pokusíme spojení zaregistrovat. Tím programově nahradíme ruční vytvoření Uživatelského DNS s požadovaným názvem a vlastnostmi, viz obr. 83. Určíme:

- cestu k požadovanému databázovému dokumentu,
- systémovou databázi s definicí uživatelských práv,
- název datového zdroje ODBC.

| Nastavení ODB | C pro Microsoft Access 97 | × | |
|---------------------------|---|--------------------|--|
| <u>N</u> ázev zdroje dat: | Lab07Sol | OK | |
| <u>P</u> opis: | Cvičná databáze | | |
| Databáze | | Storno | |
| Databáze: D:\ | Databáze: D:\\Cviceni\Lab07\Lab07Sol.mdb | | |
| Vy <u>b</u> rat | Vytvořit <u>O</u> pravit <u>Z</u> komprimovat | <u>U</u> přesnit | |
| – Systémová datab | báze | | |
| © Žá <u>d</u> ná | | | |
| O <u>a</u> tabáze: | D:\\Lab07\PROBLEM.MDW | | |
| | <u>S</u> ystémová databáze | <u>M</u> ožnosti>> | |

Obr. 83. Vlastnosti ODBC spojení

Pozn.: Výpis programu obsahuje několik pomocných funkcí. Zejména je to funkce **MakeSQLDate(Datum)**, která upraví zadané datum do formátu použitelného v SQL dotazu (mm/dd/yy), dále **FileNameSet()** a **DataPathSet()**, které pomohou najít požadované databázové dokumenty pomocí obvyklého dialogu **OpenFilename** (Otevřít).

13. Výstup dat na web (Lab10)

V současnosti je již běžnou součástí aplikací přístup k datům prostřednictvím sítě internet, obvykle pomocí www. Možností realizace takové komunikace je několik, my využijeme služeb podporovaných aplikacemi Microsoft. MS-Access nabízí:

- uložení objektů ve formátu HTML, ve statické podobě,
- vytvoření dynamických stránek HTX/IDC,
- vytvoření dynamických stránek ASP (ActiveX Server Page).

Pro nás je samozřejmě zajímavější tvorba dynamických stránek. Pro oba nabízené systémy je třeba mít k dispozici www server *Microsoft Internet Information Server*, nebo alespoň *Personal Web Server*, který je již standardní součástí Windows 98. V systému HTX/IDC je třeba vždy vytvořit dvojici dokumentů IDC s definicí SQL dotazu a HTX obsahující formátovací předpis pro převod výsledku dotazu do HTML dokumentu. Naproti tomu dokument ASP obsahuje obojí. Navíc může obsahovat více SQL dotazů, což systém HTX/IDC nedovoluje.

Naším úkolem je vytvořit www stránky umožňující zadat jméno osoby a následně zobrazit všechny její úkoly. K realizaci stránek můžeme využít *Průvodce publikováním na síť www*. K tomu je ale nutno vytvořit dotaz realizující naše požadavky (SELECT Ukoly.* FROM Ukoly WHERE Ukoly.Osoba = [Pracovnik];) a ten exportovat do ASP souboru. Průvodce vytvoří dva dokumenty. První bude obsahovat formulář pro zadání jména pracovníka a otevření druhého dokumentu. Ten provede SQL dotaz a zformátuje výsledek do HTML dokumentu.

Bohužel po umístění dokumentů na www server zjistíme, že nebudou správně pracovat. Průvodce totiž opomíjí správně omezit řetězcovou konstantu apostrofy a vložené kódování (Server.HTMLEncode) způsobuje potíže. Proto je výhodné, umět sestavit ASP dokument ručně. Je to HTML dokument doplněný o skriptovací příkazy řídící zpracování dat, včetně definice spojení s databází pomocí ODBC (Start.asp):

```
<HTML>
<HEAD>
```

```
<META HTTP-EOUIV="Content-Type"
    CONTENT="text/html;charset=windows-1250">
<TITLE>Úkoly zvoleného pracovníka</TITLE>
</HEAD>
<BODY Background="Bcgrnd.gif">
<%
Param = Request.QueryString("Param")
Data = Request.QueryString("Data")
%>
<%
If IsObject(Session("Lab07Sol_conn")) Then
    Set conn = Session("Lab07Sol conn")
Else
    Set conn =
   Server.CreateObject("ADODB.Connection")
    conn.open "Lab07Sol", "Host", ""
    Set Session("Lab07Sol_conn") = conn
End If
%>
```

<%

```
sql = "SELECT Osoby.* FROM Osoby "
   If cstr(Param) <> "" And cstr(Data) <> "" Then
        sql = sql & " And [" & cstr(Param) & "] =
   " & cstr(Data)
    End If
    Set rs =
   Server.CreateObject("ADODB.Recordset")
   rs.Open sql, conn, 3, 3
%>
<H1 Align=CENTER>Úkoly pro pracovníka:</H1>
<FORM NAME="Pracovnik" METHOD="GET"
   ACTION="Ukoly.asp">
<SELECT NAME="[Pracovnik]">
<%
On Error Resume Next
do while Not rs.eof
%>
    <OPTION
   VALUE="<%=rs.Fields("Osoba").Value%>"><%=rs.Fi
   elds("Osoba").Value%>
<%
rs.MoveNext
loop%>
</SELECT>
<INPUT TYPE="Submit" VALUE="Spustit dotaz">
</FORM>
<HR>
<%
   sql = "SELECT Problemy.* FROM Problemy "
   If cstr(Param) <> "" And cstr(Data) <> "" Then
        sql = sql & " And [" & cstr(Param) & "] =
   " & cstr(Data)
    End If
```

```
Set rs =
    Server.CreateObject("ADODB.Recordset")
    rs.Open sql, conn, 3, 3
%>
<H1 Align=CENTER>Úkoly pro správce problému:</H1>
<FORM NAME="Vedouci" METHOD="GET"
    ACTION="Ukoly.asp">
<SELECT NAME="[Pracovnik]">
<%
On Error Resume Next
do while Not rs.eof
%>
    <OPTION
    VALUE="<%=rs.Fields("Osoba").Value%>"><%=rs.Fi
    elds("Nazev").Value%>
<%
rs.MoveNext
<%gool
</SELECT>
<INPUT TYPE="Submit" VALUE="Spustit dotaz">
</FORM>
</BODY>
</HTMI_{i}>
```

Zobrazená HTML stránka nabízí výběr osoby ze seznamu osob, nebo podle odpovědnosti za konkrétní úkol, viz obr. 84. Po určení osoby je spuštěn ASP dokument Ukoly.asp, který zpracuje úkoly požadované osoby a zobrazí, viz obr. 85. Skriptovací jazyk přitom umožňuje zjistit, zda osoba má evidovány úkoly a podle toho se rozhodne o způsobu sestavení HTML dokumentu

```
<HTMI>
<HEAD>
<META HTTP-EQUIV="Content-Type"
   CONTENT="text/html;charset=windows-
    1250">
<TITLE>Úkoly zvoleného pracovníka</TITLE>
</HEAD>
<BODY Background="Bcgrnd.gif">
<%
Param = Request.QueryString("Param")
Data = Request.QueryString("Data")
응>
<%
If IsObject(Session("Lab07Sol conn"))
    Then
   Set conn = Session("Lab07Sol conn")
Else
    Set conn =
   Server.CreateObject("ADODB.Connection
    ")
   conn.open "Lab07Sol", "Host", ""
   Set Session("Lab07Sol conn") = conn
End If
%>
<%
   sql = "SELECT Ukoly.* FROM Ukoly WHERE
   (((Ukoly.Osoba)='" &
   Request.QueryString("[Pracovnik]") & "')) "
   If cstr(Param) <> "" And cstr(Data) <> "" Then
        sql = sql & " And [" & cstr(Param) & "] =
    " & cstr(Data)
    End If
    Set rs =
   Server.CreateObject("ADODB.Recordset")
```

| 🖞 Úkoly zvoleného pracovníka - Microsoft Internet Explorer 📃 🔲 🗙 |
|---|
| Soubor Úpr <u>avy Z</u> obrazit <u>P</u> řejít <u>O</u> blíbené Nápo <u>v</u> ěda |
| $(\div \cdot \rightarrow \cdot \otimes \boxtimes \bigtriangleup \otimes \boxtimes \boxtimes \otimes \boxtimes \boxtimes \boxtimes $ |
| Adresa 🔄 http://localhost/books/cviceni/start.asp 🔄 🗍 Odkazy |
| Úkoly pro pracovníka: |
| Kotrba, Matěj Spustit dotaz |
| Úkoly pro správce problému: |
| Prodej databázové podpory systému řešení problémů 💌 Spustit dotaz |
| |
| 🖹 Hotovo 🛛 👘 👘 Místní zóna vnitřní sítě intranet 🥢 |
| Obr. 84. HTML stránka se zadáním jména osoby |
| rs.Open sql, conn, 3, 3 %> |

```
<H1 Align=CENTER>Úkoly pro pracovníka:</H1>
<H2
```

Align=CENTER><%=Request.QueryString("[Pracovni kl")%></H2> <TABLE BORDER=1 WIDTH="100%">

```
<TR><TH ALIGN=CENTER VALIGN=TOP
```

WIDTH="20%">Zahájení</TH>

```
<TH ALIGN=CENTER VALIGN=TOP
```

```
WIDTH="20%">Ukončení</TH>
```

```
<TH ALIGN=CENTER VALIGN=TOP WIDTH="55%">Popis
    úkolu</TH>
    <TH ALIGN=CENTER VALIGN=TOP WIDTH="5%">ID</TH>
</TR>
<%
On Error Resume Next
If rs.EOF then
%>
<H1 align=CENTER>Žádné úkoly nejsou k
    dispozici</H1>
<%
else
    rs.MoveFirst
   do while Not rs.eof
%>
    <TR>
    <TD ALIGN=CENTER VALIGN=TOP
    WIDTH="20%"><%=rs.Fields("DatumOd").Value</pre>
    %></TD>
    <TD ALIGN=CENTER VALIGN=TOP
    WIDTH="20%"><%=rs.Fields("DatumDo").Value
    %></TD>
    <TD ALTGN=LEFT VALTGN=TOP
    WIDTH="55%"><%=rs.Fields("Reseni").Value%</pre>
    ></TD>
    <TD ALIGN=CENTER VALIGN=TOP
    WIDTH="5%"><%=rs.Fields("ID").Value%></TD
    >
    </TR>
<%
    rs.MoveNext
   loop
end if
%>
```

```
</TABLE>
<HR>
<P ALIGN=CENTER><A HREF="Start.asp">Návrat k
výběru osob</A>
</BODY>
</HTML>
```

| 🛞 | 2 🖧 🔕 | 🗈 🥝 🍳 🖬 🖏 🎒 🖆 | | | | |
|--|--|---|----------------------------|--|--|--|
| sa 🛃 http://loca | alhost/books/cviceni/ | /Ukoly.asp?%5BPracovnik%5D=Kotrba%2C+Mat%ECj | _](| | | |
| Kotrba, Matěj | | | | | | |
| | | | | | | |
| Zahájení | Ukončení | Popis úkolu | D | | | |
| Zahájení 23.3.1999 | Ukončení 6.4.1999 | Popis úkolu Tvorba tabulek | ID 1 | | | |
| Zahájení 23.3.1999 23.3.1999 | Ukončení 6.4.1999 24.3.1999 | Popis úkolu Tvorba tabulek Datová analýza | 1 2 | | | |
| Zahájení 23.3.1999 23.3.1999 23.3.1999 | Ukončení 6.4.1999 24.3.1999 25.3.1999 | Popis úkolu Tvorba tabulek Datová analýza Analýza objektů | 1 2 3 | | | |
| Zahájení 23.3.1999 23.3.1999 23.3.1999 23.3.1999 24.3.1999 | Ukončení 6.4.1999 24.3.1999 25.3.1999 27.3.1999 | Popis úkolu Tvorba tabulek Datová analýza Analýza objektů Analýza vazeb mezi objekty | 1 2 3 4 | | | |
| Zahájení 23.3.1999 23.3.1999 23.3.1999 23.3.1999 24.3.1999 27.3.1999 | Ukončení 6.4.1999 24.3.1999 25.3.1999 27.3.1999 30.3.1999 | Popis úkolu Tvorba tabulek Datová analýza Analýza objektů Analýza vazeb mezi objekty Optimalizace rozvržení dat | 1 1 2 3 4 5 | | | |

Obr. 85. HTML stránka s úkoly vybrané osoby

Literatura

- BEJČEK, VLASTIMIL 1992. *Databázové systémy*. 1. vyd. Brno : VUT, 1992, 218 s. ISBN 80-214-0422-1.
- BÍLEK, MARTIN 1995. MS Access 2.0 pro Windows. 1. vyd. Praha: Grada Publishing, 1995, 164 s. ISBN 80-7169-185-2
- BILLINGS, SCOTT, RHEMANN, JOE aj 1997. Access 97 Programming. 1. vyd. Indianapolis (USA): Sams Publishing, 1997, 874 s. ISBN 0-672-31049-X.
- BRODSKÝ, JAN, STAUDEK, JAN & POKORNÝ, JAROSLAV 1992. *Operační a databázové systémy*. 1. vyd. Brno: VUT, 1992, 162 s. ISBN 80-214-0444-2.
- CARLBERG CONRAD aj. 1996. MS Office 95 Access, PowerPoint, Schedule+ Profesionální řešení. 1. vyd. Brno : UNIS Publishing, 1996, 432 s. ISBN 0-7897-0391-2.
- COMPUTER WORLD. *Databáze*. *Série článků* [online]. Computer World. Cit. 9. 2. 1999. Dostupný z www <URL: http://www.cw.cz/db/index.htm>.
- CORNELL, GARY 1999. *Microsoft Visual Basic Scripting*. *Příručka programátora*. 1. vyd. Praha : Computer Press, 1999, 278 s. ISBN 80-7226-144-4.

- DOBDA, LUKÁŠ 1998. Ochrana dat v informačních systémech. 1. vyd. Praha : Grada Publishing, 1998, 288 s. ISBN 80-7169-479-7.
- FARANA, RADIM & VOJÁČEK, MICHAL 1993. Databázové systémy. Microsoft Access. 1. vyd. Ostrava : KAKI/kat.ATŘ VŠB Ostrava, 1993, 119 s.
- FARANA, RADIM 1995. Aplikace počítačů v řízení. Relační databáze. 1. vyd. Ostrava : KAKI, 1995. 129 s. ISBN 80-02-01042-6.
- FARANA, RADIM 1996. Řešené problémy v databázi MS-Access. Interní učební text. 1. vyd. Ostrava : kat. ATŘ VŠB-TU Ostrava, 1996. 60 s.
- FIKÁČEK, IVO, ROZEHNAL, IVO & FIKÁČEK, MARTIN 1997. Microsoft Access 97 snadno a rychle. 1 vyd. Praha : Grada Publishing, 1997, 144 s. ISBN 80-7169-529-7.
- FIKÁČEK, IVO, ROZEHNAL, IVO 1996. *Access 7 pro Windows 95*. 1. vyd. Praha : Grada, 132 s. ISBN 80-7169-290-5.
- KRÁL, JAROSLAV 1998. *Informační systémy*. 1. vyd. Veletiny : Science, 1998, 360 s. ISBN 80-86083-00-4.

- KREJČÍ, RICHARD 1997. Tvorba www stránek v Office 97 a Office 7 snadno a rychle. 1. vyd. Praha: Grada Publishing, 1997, 199 s. ISBN 80-7169-482-7.
- MICROSOFT CORPORATION 1996a. *Microsoft TechNet. Technical Information Network* [CD ROM]. February 1996, Volume 4, Issue 2.
- MICROSOFT CORPORATION 1996b. Mastering Microsoft Access Programming. Interactive Training [CD ROM]. Microsoft Press, 1996.
- Microsoft Office 97 Resource Kit. 1. vyd. Praha : Computer Press, 1997, 1300 s. ISBN 80-7226-017-0.
- Microsoft Office 97 Visual Basic příručka programátora. 1. vyd. Praha : Computer Press, 1998, 482 s. ISBN 80-7226-117-7.
- MOLNÁR, ZDENĚK 1992. *Moderní metody řízení informačních systémů*. 1. vyd. Praha : Grada, 1992, 352 s. ISBN 80-85623-07-2.
- MOOS PETR 1993. *Informační technologie*. 1. vyd. Praha : Vydavatelství ČVUT, 1993, 220 s. ISBN 80-01-01048-1.
- MORKES, DAVID 1998. Access 97 CZ. 1. vyd. Praha : Computer Press, 1998. ISBN 80-7226-053-7.
- MULLEN, ROBERT aj. 1997. *Microsoft Office 97 Profesional*. 1. vyd. Brno: UNIS Publishing, 1997, 336 s. ISBN 80-86097-07-2.

- NESRSTA, LADISLAV & MALÝ, VLASTIMIL 1993. *Microsoft* Access 1.0. 1. vyd. Brno : TH' Systems a. s., 1993, 96 s.
- PLECHÁČ, VLADIMÍR & HERNADY, ROBERT 1995. Access 2.0. Tipy, triky v příkladech. 1. vyd. Praha : GComp, 1995, 254 s. ISBN 80-85649-34-9.
- PLECHÁČ, VLADIMÍR & SEMETKOVSKÝ, LADISLAV 1996. *Klient/server ODBC*. 1. vyd. Praha : GComp, 1996, 116 s. ISBN 80-85649-41-1.
- PLECHÁČ, VLADIMÍR 1993. MS Access. Popis prostředí. 1 vyd. Praha : GComp, 1993, 136 s. ISBN 80-85649-12-8.
- PLECHÁČ, VLADIMÍR 1994. Access 2.0. Učebnice programování. 1. vyd. Praha : GComp, 1994, 224 s. ISBN 80-85649-23-3.
- PLECHÁČ, VLADIMÍR 1995. *Access 2.0. Průvodce systémem.* 2. vyd. Praha : GComp, 1995, 320 s. ISBN 80-85649-28-4.
- PLECHÁČ, VLADIMÍR 1996. Access 7.0. Kompletní průvodce. 1. vyd. Praha : GComp, 411 s. ISBN 80-85649-53-5.
- PLECHÁČ, VLADIMÍR 1997. *Od klient/server k Intranetu*. 1. vyd. Praha : GComp, 1997, 144 s. ISBN 80-85649-55-1.
- Рокопич, JAN 1993. *Učíme se SQL*. 1. vyd. Praha : Plus, s. r. o., 1993, 568 s. ISBN 80-85297-47-7.
- Рокогиу́, JAN 1997a. *Microsoft Office 97 a internet*. 1. vyd. Č. Budějovice : Kopp naklad., 1997, 196 s. ISBN 80-85828-93-6.

- Рокоrný, Jan 1997b. *Spolupráce aplikací MS Office*. 1. vyd. Č. Budějovice : KOPP, 1997, 204 s. ISBN 80-85828-81-2
- Рокопич, JAN 1997с. *Basic pro aplikace Accessu* 97. 1. vyd. Č. Budějovice : KOPP, 1997, 220 s. ISBN 80-7232-003-3.
- POKORNÝ, JAROSLAV & HALAŠKA, IVAN 1995. Databázové systémy. Cvičení (jazyk SQL a systém ORACLE). 1. vyd. Praha : ČVUT, 1995, 244 s. ISBN 80-01-01382-0.
- POKORNÝ, JAROSLAV & HALAŠKA, IVAN 1998. *Databázové systémy*. 1. vyd. Praha : ČVUT, 1998, 146 s. ISBN 80-01-01724-9.
- POKORNÝ, JAROSLAV 1992. Databázové systémy a jejich použití v informačních systémech. 1. vyd. Praha : Academia, 1992, 320 s. ISBN 80-200-0177-8.
- POKORNÝ, JAROSLAV 1994. *Dotazovací jazyky*. 1. vyd. Veletiny : SCIENCE, 1994, 228 s. ISBN 80-901475-2-6.
- POKORNÝ, JAROSLAV 1998. *Databázová abeceda*. 1. vyd. Veletiny : Science, 1998, 260 s. ISBN 80-86083-0-20.
- ROMAN, STEVEN 1999. Microsoft Access. Návrh a programování databází : co potřebujete opravdu vědět o tvorbě databází. 1. vyd. Praha : Computer Press, 1999, 250 s. ISBN 80-7226-133-9.
- SALEMI, JOE 1993. *Databáze klient/server*. *Průvodce*. 1. vyd. Brno : Unis Publishing, 1993, 273 s.

- SERVER FS VŠB-TU OSTRAVA. Informační systém Fakutty strojní Vysoké školy báňské - Technické univerzity Ostrava [online]. Ostrava: FS VŠB-TUO. Cit. 9. 2. 1999. Dostupný z www <URL: http://www.fs.vsb.cz/>.
- SIMPSON, ALAN & OLSON, ELIZABETH 1998. *Access* 97. 1. vyd. Praha : Grada Publishing, 1998, 968 s. ISBN 80-7169-612-9.
- SOLOMON, CHRISTINE 1998. Tvorba aplikací v Microsoft office 97 pomocí jazyka Visual Basic. 1. vyd. Praha : Computer Press, 1998, 598 s. ISBN 80-7226-025-1.
- SOUKUP, RON 1998. *Mistrovství v SQL Serveru 6.5.* 1. vyd. Praha: Computer Press, 1998, 850 s. ISBN 80-7226-092-8.
- STEINER, JOSEF & VALENTIN, ROBERT 1998. Microsoft Access 97. Kompletní kapesní průvodce. 1. vyd. Praha: Grada, 1998, 351 s. ISBN 80-7169-618-8.
- STEINER, JOSEF & VALENTIN, ROBERT 1998. Access 97. Kompletní kapesní průvodce. Praha : Grada Publishing, 1998, 352 s. ISBN 80-7169-618-8.
- STRAKA, MIROSLAV 1992. *Vývoj databázových aplikaci*.1. vyd. Praha : Grada, 1992, 160 s. ISBN 80-85424-43-6.
- ŠIMEK, TOMÁŠ. *Microsoft Access 2.0 CZ: Základní příručka uživatele*. 1. vyd. Praha : Computer Press, b.r., 148 s. ISBN 80-85896-10-9.

- ŠIMŮNEK, MILAN 1999. SQL. Kompletní kapesní průvodce. 1. vyd. Praha : Grada Publishing, 1999, 248 s. ISBN 80-7169-692-7.
- TSICHRITZIS DIONYSIOS C. & LOCHOVSKY, FREDERIK H. 1987. Databázové systémy. 1 vyd. Praha : SNTL, 1987, 384 s.
- VARNER, DAVID & LEITGEB, IVO 1997. Access 7 efektivně. 1. vyd. Brno : CCB, 1997, 218 s. ISBN 80-85825-21-X.
- VAUGHN, WILLIAM R. 1998. Visual Basic pro SQL Server. 1. vyd. Praha : Computer Press, 1998, 962 s. ISBN 80-85896-99-0.
- VIESCAS, JOHN L. 1997. Microsoft Access pro Windows 95 odborný průvodce. 1. vyd. Praha : Computer Press. 1997, 822 s. ISBN 80-85896-80-X.
- VIESCAS, JOHN L. 1998. Mistrovství v Microsoft Access 97. 1. vyd. Praha: Computer Press. 1998, 750 s. ISBN 80-7226-107-X.
- VLČEK, JAROSLAV 1994. *Inženýrská informatika*. 1. vyd. Praha : Vydavatelství ČVUT, 1994, 281 s. ISBN 80-01-01071-6.

| Číslo skladové: | 1940 | 300 | |
|--|--|-----|--|
| Určeno pro posluchače: | 3. ročník Bc., 4. ročník Ing. FS | | |
| Autor: | Ing. Radim Farana, CSc. | | |
| Katedra, institut: | automatizační techniky a řízení | 352 | |
| Název: | 7: Tvorba relačních databázových systémů | | |
| Místo, rok, vydání: | Ostrava, 1999, 1. vydání | | |
| Počet stran: | 100 | | |
| Vydala: | VŠB – TECHNICKÁ UNIVERZITA OSTRAVA | | |
| Tisk: | REPRONIS Jiří Němec Nádražní 93, 701 00 Ostrava 1 | | |
| Náklad: | 200 ks | | |
| Tématická skupina: | 17 | | |
| Povoleno MK ČSR č. j. 21.514/79 ze dne 4. 12. 1979 | | | |

ISBN 80-7078-706-6